Master's Thesis

Raul Alejandro Morquecho Martinez

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.
Espoo 31.3.2016

Thesis supervisor:

Prof. Tarik Taleb

Thesis advisors:

M.Sc. Wolfram Roth

M.Sc. Jan Fichtmueller

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Raul Alejandro Morquecho Martinez

Title: Delivery of encryption keys in TETRA networks

Date: 31.3.2016 Language: English Number of pages: 7 + 70

Department of Communications and Networking

Professorship: Prof. Tarik Taleb

Supervisor: Prof. Tarik Taleb

Advisors: M.Sc. Wolfram Roth, M.Sc. Jan Fichtmueller

Professional mobile radio networks, to which TETRA belongs, are used mostly by public safety departments, which are in nature non-commercial.
This thesis was produced for the company Secure Land Communications. Due to our customers' needs for a mechanism to distribute their communication's encryption keys over the air, the End-to-End Encryption project with Over The Air Keying (OTAK) feature was created. Our job was to test the behavior of a network when extra load due to this new solution is introduced, as well as discover any defects in order to make the proper corrections. In order to do this, we defined a test strategy and test plan including a traffic load profile, a suitable test environment and test cases focusing on high loads of OTAK traffic, so we would identify any important errors, submit them for correction, and test the solutions of those corrections.
Our verification yielded the validation of the correct operation of features and many defect discoveries, some of them very critical, which were corrected to have as a result a significant improvement and a better solution which benefits our company's portfolio. Not everything, however, has been solved at the moment of finishing this thesis, there are still issues and some new errors were introduced as part of the introduced fixes, these are expected to be solved during subsequent releases of the solution, together with minor defects not corrected during this first phase.

Keywords: encryption keys, security, TETRA, OTAK, PMR.

Preface

I want to thank all the people who made possible this thesis, my instructors Wolfram Roth and Jan Fichtmueller, who gave me the opportunity to participate in this project, my professor Tarik Taleb, for his guidance, and all the people I worked with during this project, without whom it would have been impossible to accomplish the results of this thesis work.

I would also like to thank my family, who always supported me morally during the duration of my studies abroad, my mother, to whom I owe my life, and especially, I would like to dedicate this thesis and degree to the memory of my father, who left this world during the time of my studies in Finland, and who will always be remembered warmly in my heart.

Espoo, 31.3.2016

Raul Morquecho

Table of Contents

Symbols and Abbreviations

ATCA	= Advanced Telecommunications Computing Architecture
OTAK	= Over The Air Keying
E2EE	= End to End Encryption
KMF	= Key Management Facility
DXTA	= Digital eXchange for TETRA type ATCA
TCS	= TETRA Connectivity Server
API	= Application Programming Interface
SCT	= Smart Card Tool
KSCC	= Key management facility Smart Card Configurator
SLC	= Secured Land Communications
SIM	= Subscriber Identity Module
CG	= Crypto Group
TBS	= TETRA Base Station
TR	= TETRA Radio
TEK	= Traffic Encryption Keys
OOB	= Out Of Band
KEK	= Key Encryption Key
SEK	= Signaling Encryption Key
SwMI	= Switching and Management Infrastructure
SDS	= Short Data Service
FEK	= File Encryption Key
BEK	= Backup Encryption Key
IEK	= Internal Encryption Key
RCS	= Radio Console System
RecGw	= Recording Gateway
MGEM	= Multi Generic Encryption Module
SCM	= Smart Card Module
SC4KMF	= Smart Card for Key Management Facility
UDT	= User Domain Tool
ISSI	= Individual Short Subscriber Identity
CDD	= Configuration and Data Distribution server
UL	= UpLink
DL	= DownLink
ACK	= ACKnowledgement
NIC	= Network Interface Card
GMT	= Greenwich Mean Time
ISI	= Inter Systems Interface
G4WI	= Generic 4 Wire Interface

BTS	= Base Transceiver Station
MS	= Mobile Subscriber
PTT	= Push To Talk
DMO	= Direct Mode Operation
PSTN	= Public Switched Telephone Network
TETRA	= TErrestrial Trunked RAdio
ETSI	= European Telecommunications Standards Institute
TEDS	= TETRA Enhanced Data Services
TDMA	= Time Division Multiple Access
MCCH	= Main Control CHannel
TCH	= Traffic CHannel
SCCH	= Secondary Control CHannel
TCCA	= TETRA and Critical Communications Association
SFPG	= Security and Fraud Prevention Group
LTE	= Long Term Evolution
GSSI	= Group Short Subscriber Identity

Symbols

~	Approximately

1. Introduction

Professional Mobile Radio (PMR) has been around for many years, it started last century with the first analog and hugely sized portable radios which used one communication channel per assigned frequency band. Nowadays, PMR has evolved and is completely digital, it uses time division multiplexing and different coding techniques in order to make the best possible usage of the radio spectrum assigned to it.

TErrestrial Trunked RAdio (TETRA), as specified by ETSI (2016), is a flavor of PMR designed by the European Telecommunications Standards Institute. TETRA is meant mostly for safety and security organizations, and used also by public transport, gas, oil and mining plants; it brings specific features critical to high availability and secure radio communications to customers in need of an own dedicated secure cellular network.

One of TETRA's features is the use of encrypted voice and data, which is available by using encryption keys in sender and receiver sides, these encryption keys can be exchanged by different mechanisms. For the standardization of these mechanisms, ETSI works together with the TETRA and Critical Communications Association (TCCA), which in turn, delegates the task to a sub-group called the Security and Fraud Prevention Group (SFPG), as specified by TCCA (2016).

The SFPG defines a method to deliver encryption keys to all users in a TETRA network wirelessly without the need to take physically the TETRA Radios (TR) to the maintenance center every time there is a need to change encryption keys. This mechanism is called Over The Air Keying (OTAK).

Prior to the project described in this thesis, in Secure Land Communications, such a mechanism did not exist for large-scale networks. For End-to-End Encryption (E2EE), either out of band delivery of keys (manual configuration) or a downgraded version of the OTAK mechanism which supported only the delivery of encryption keys to a few hundred users was in use.
Once this solution using OTAK is taken into use, the amount of users served by this new solution will be on the range of thousands, and can be expanded to hundreds of thousands by adding several servers together.

This thesis addresses the process of testing on a system level the implementation of the E2EE solution using OTAK, where all or most of the critical components interact together to perform a network traffic similar to that of the customer.

This thesis is only a small portion of the work done to implement this solution, it neither addresses the development of the solution's components nor the actual implementation of the solution in field. It addresses, however, the verification of the solution to find defects and follow their correction, so the end result will be a mature solution capable of implementing the required functionality without introducing unwanted side effects.

2. Scope of the project

This project was created to fulfill one of our customer’s needs, the delivery of keys over the air from the keys' repository center to the mobile radios deployed in field without the need to manually reconfigure the set of encryption keys every time they are renewed.

The project is quite extensive, it comprises development of new tools in 2 sites (Jyväskylä in Finland and Paris in France), testing in 2 sites (Paris in France and Helsinki in Finland), as well as the deployment of the solution in the customers network.

Due to the extent of the project, the part concerning Helsinki, in which I am involved, is just a small portion compared to the overall project. Due to the nature of this diversification of work, I can define the scope of the project in the Helsinki site as the following:

The scope for our department is to test over the air interface (base station to radio handheld), the behavior of the radio spectrum capacity when a significant amount of extra load is applied due to the new signaling traffic introduced with the new solution in the area covered by one Tetra Base Station (TBS).
Above that, the scope of the project covers the discovery of any defects in the solution in terms of the tested capacity in order to make the proper corrections before deploying them into the customer's network.

3. Background Information

3.1. Professional Mobile Radio (PMR) systems

As defined by Ketterling, H. (2003) "PMR offers two-way radio communication carrying speech, data or a mix of both in non-public networks tailored to the specific operational needs of professional mobile user groups for efficient and flexible communication within their area of daily operation."

As the definition specifies, these networks are non-public, mainly because the customers using PMR are those who do not want to use commercial operators due to security concerns; the reason for this is because PMR networks are widely used by safety and security organisms belonging to a specific non-commercial organization, such as a police department of a given government.

Besides of the security concerns specified above, Hans Ketterling (2003) also specifies in his book the requirements for what an organization usually choses PMR as its main means of communication, I list in the paragraph below the main attributes according to him of PMR as well as a brief explanation of the them:

- Group, fleet and broadcast calls: Groups and other short dialing numbers can be programmed in the radios in order to have separate channels to use depending of what users we want to contact.
- Dispatcher users: In PMR networks, there are usually users which orchestrate communications or operational management functions, these are called dispatchers.
- Access to other communication's systems: from PMR networks, if it is allowed in the configuration, the users can also communicate to other systems such as PSTN, commercial cellular systems, internet, etc.
- Voice and data encryption over the air and End-to-End: this is accomplished by different means; this thesis explains the way to accomplish End-to-End encryption; Air Interface encryption is not in the scope of this thesis.
- Enabling/Disabling of radio terminals and certain modes of operations such as ambience listening which is a feature in which the microphone of the terminal can be activated remotely, like in the case of a stolen radio, or when the user cannot do any actions manually, such as a hostage situation.
- Heavy duty equipment: users of PMR often operate in dangerous environments in which regular mobile phones cannot be used, such as chemical plants or where there are fires (fire department personnel), for that reason the radios must be designed to operate in extreme environments.

The previous list of features is by no means a comprehensive list, however, it lists the most important points and explains why regular cellular telephony/equipment is not used by the organizations which chose to communicate through PMR.

3.1.1. What is TETRA?

The Terrestrial Trunked Radio is a standard defined by the European Telecommunications Standards Institute, which is the organization in charge to publish telecommunication standards to be used mandatorily in Europe.

According to SELEX communications (2007), TETRA is the first truly digital PMR standard, before TETRA, the standard in use was called MPT1327, however, this standard was analogue, and lacked all the characteristics of digital communications.
Even though TETRA was created by ETSI, it has gained worldwide approval and it is used in many other parts of the world outside of Europe, where it is not mandatory, but has been adopted for emergency and security communication networks.

TETRA can be seen as a specific layer over the general PMR technology, adding services specifically designed for public safety organizations.
As mentioned by ETSI (2016), there are unique PMR services in the TETRA standard, such as:

- Fast access to calls and short transmission delays: fast access to calls is usually performed by a Push To Talk (PTT) button which gives a direct line to a pre-selected talk group.
- Direct Mode Operation (DMO): PMR mobile stations can operate directly with one another (given the fact that they are on a determined distance range) without the need of a base station.
- High level encryption (Air Interface and E2EE with different algorithms)
- Priority calls: the user can for example make an emergency call which could take network resources already in use by a less important call.
- High speed data communication with TETRA Enhanced Data Services (TEDS, not to be covered in this thesis).

TETRA uses Time Division Multiple Access (TDMA) to allocate resources in the Air Interface (AI), every TBS works with one Main Control Channel (MCCH), which has the responsibility of managing the signaling, while the Traffic Channels (TCH) carries the voice or data accordingly.
In TETRA, a TBS can be configured accordingly with a predefined amount of Secondary Control Channels (SCCH), which serve as aid channels to the MCCH in case of high traffic of signaling on that specific TBS.

TETRA is at the moment suffering a transformation from purely digital PMR to hybrid communications due to the new range of technologies available (such as LTE), which do not support the specific requirements of TETRA, but when using it with TETRA, the combination allows a higher data throughput while still making use of TETRA specific advantages, such as group and emergency communications.

3.1.2. TETRA systems

As described by Secure Land Communications (2015), TETRA products range from infrastructure comprising traffic switches/exchanges and base stations, passing through end user hardware like handheld radios and dispatcher work stations, going all the way to software applications such as packages for managing radio subscribers or for viewing real time tracking of the radios deployed in field.

SLC develops, tests, deploys and maintains TETRA core hardware and software products such as exchanges and base stations. Other smaller products are also developed and tested such as the TETRA Connectivity Server (TCS), which serves as an interface between the TETRA system and third party applications by implementing an Application Programming Interface (API) capable of receiving and delivering commands for interaction between third party equipment and the TETRA backbone.

For the project related to this thesis, I only focused in a very small part of the system, that is, the distribution of keys for encryption of voice and short data message traffic; this traffic is distributed through the network using a protocol called Over The Air Keying (OTAK). In SLC's implementation, OTAK End-to-End Encryption's backbone is the Key Management Facility (KMF); the function of this KMF is to keep track in its database of the keys for encrypting communication, and coordinate the use of these by loading them over the air to the TRs which have been registered to it and which support delivery of encryption keys over the air.

For our specific case, it is worth to notice that the core of the solution delivered by this end-to-end encryption project, the KMF, acts like a third party application in order to achieve connectivity with the TETRA network through the TCS API for delivery of encryption keys and other signaling.
The KMF product itself, is developed in France, however, part of the testing was moved to Helsinki.

In Finland also the development of other TETRA tools important for our project takes place, such as the Smart Card Tool (SCT) and KMF Smart Card Configurator (KSCC), which I will describe more in detail in sections 4.4.4.2 and 4.4.1.2 respectively.

For a short description of the specific devices used in this project, see section 4.4 in this document.

3.2. Integration & Verification

"Integration is the process of combining all the unit-tested program units into a complete system." (Bishop, 2008, p. 322).
What Bishop (2008) described in that sentence of his book is exactly what we did in this project.

In our case, due to an extensive list of sub-systems involved in the E2EE solution, integration and verification is divided in different areas, tested in geographically separate places. In Helsinki, we test 4

units in our system verification, these units are: DXT, TBS, TH1n (TETRA Radio) and KMF. Other units, such as dispatcher applications or SCT (application to program smart cards) are tested in France.

The main point of our tests is to verify, once integrated, that the 4 units assigned to us work as desired, without interfering one with the operations of the other. On top of this, our tests in Helsinki are focused specifically on checking the effect that the E2EE solution using OTAK will cause over the AI.

3.3. Cyber security

3.3.1. Symmetric Encryption

There are two kinds of encryption techniques, symmetric and asymmetric. For our purposes, we focus on symmetric encryption, because this project's solution based on TETRA systems uses symmetric encryption to encrypt/decrypt the voice calls between users.
We should know though, that symmetric encryption uses only one key to encrypt and decrypt the data, that makes it simple to operate, however, the challenge is to distribute the keys to all the nodes in a secure way, that is why the KMF solution was created.

As mentioned by Stallings and Brown (2008), a symmetric encryption system has five ingredients:

1. We have the plaintext, which is our original message, readable in clear
2. And we have the secret key, which is used by the encryption algorithm to create
3. The cyphertext, this is nothing more than a transposition and permutation of the plaintext by
4. The encryption algorithm, this one only dictates the basics of how to perform the permutations and transpositions, but the exact operations are indicated by the secret key.
5. The fifth element, the decryption algorithm, is nothing more than the encryption algorithm run in the reverse order.

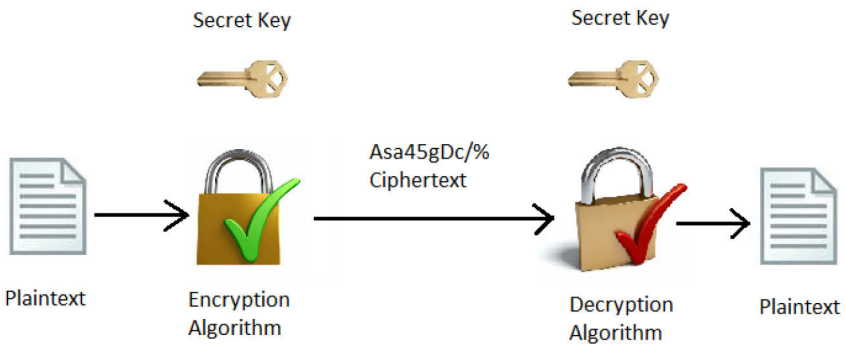

Figure 1 - Illustration of the 5 elements in an encrypted communication

3.3.2. General security concepts in TETRA systems

TETRA systems are rich in security features, according to Finland's Airbus Defence and Space (2015, d) there are, widely speaking, 5 areas where security can be seen in one way or another in TETRA networks.

Area one: Authentication.
If the network is configured for authentication, a given user must log into the network by performing an authentication mechanism no matter what kind of device he's using (a TR or a dispatcher station).

Area two: Security features.
Several measures can be taken to ensure security from an operational point of view, such measures could be for example ambience listening for hostage related scenarios, disable/enable functions in case a TR has been stolen, limited rights to dispatchers to allow them to use only certain functions, or recording of calls for audit and lawful interception.

Area three: Protecting our traffic against outsiders.
Three measures can be taken against this problem. The First and most important security measure in traffic protection is end to end encryption, which provides security from one end point in the system to the other and is the subject of this thesis. The second measure is the encryption of radio links between TRs and TBSs (explained shortly in section 3.3.2.4). And the third measure is to protect the links between the DXTs and TBSs with IPsec (not covered in this thesis).

Area four: Management.
Security in a network concerns also its management, in SLC's implementation of TETRA systems, the devices can be managed remotely by secured connections (SSH and SFTP), and the actions of users can also be logged into an audit server.

Area five: Denial of service.
Even though this does not cause stealing of data, it would cause a disaster if it is performed, to countermeasure this, base stations are prepared with jamming detection techniques to avoid this problem.

Later in this thesis, I will explain authentication and differentiate the types of encryption in a bit more detail.

3.3.2.1. Authentication

Even though authentication is not a feature which would be modified and/or tested during this project, a quick overview of it is necessary in order to have a general knowledge of the security features available in TETRA networks, especially because this feature is already in use in our customer's network.

As described in Finland's Cassidian (2013), "Authentication is a function which allows the infrastructure to check that a mobile subscriber or a workstation user is valid to operate in the system."

The document of Cassidian (2013) from Finland, states the types of authentication used in TETRA systems, we can summarize them as 3 types: Authentication of a mobile subscriber, authentication of a workstation user and authentication key distribution.

For convenience, and because it is the most relevant type of authentication for our purposes, I will shortly describe the authentication of a mobile subscriber.

In the SLC systems, the authentication of a mobile subscriber is mutual, that means that first the mobile subscriber must authenticate to the network, and after that, the mobile subscriber requests the network to authenticate to him as well. This is specified in Finnish Cassidian (2013, page 15), together with the authentication procedure explained below.

In a simplified manner, the authentication procedure is as follows:

1) The mobile subscriber registers to the network and sends a location update.
2) The network sends to the subscriber a challenge based in 3 factors: the authentication algorithm, random numbers, and the authentication key.
3) The mobile subscriber calculates the response to the challenge based on the authentication key that has been previously saved in its SIM card, and sends it back to the network.
4) Once the network validates the response of the subscriber, it allows it access to the network.

For this to happen, subscriber specific data (authentication key and Individual Tetra Subscriber Identity) must be programmed or delivered to the mobile subscriber beforehand in a secure manner, these are saved in the mobile subscriber's equipment (either on a SIM card or in the radio itself).
A third element is necessary during the authentication: the authentication algorithm, this however, can be delivered by the network while it provides the challenge to be answered by the mobile subscriber.

3.3.2.2. The concept of a Crypto Group

In TETRA systems, a Crypto Group (CG) is an entity which holds the relationship between Traffic Encryption Keys (TEKs, explained in section 4.3.1) and a range of subscribers' numbers in the network.

In TETRA systems, all radio equipment is configured with an Individual Short Subscriber Identity (ISSI), which is the personal phone number of that radio; these ISSIs are then assigned to a User Group (UG), which is the bridge between the ISSIs and the CGs. Once a UG has a range of ISSIs belonging to it, it can also be assigned a range of CGs, and the TEKs contained in those CGs will be delivered to the range of ISSIs.

This concept is bit difficult to understand only with words, to make it clearer; I give an example of this mechanism in figure 2.

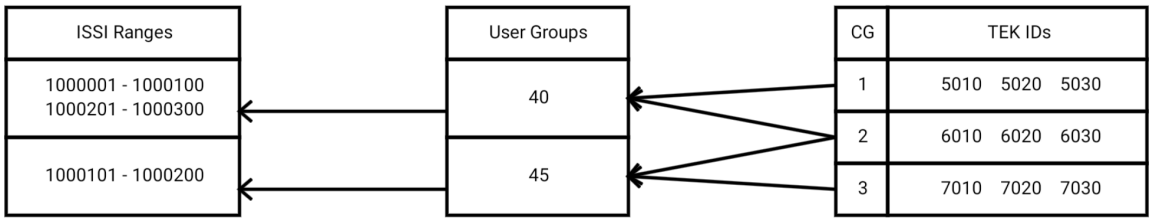

Figure 2 - Mapping of Traffic Encryption keys to Individual Short Subscriber Identities

In the example represented in figure 2, we can see 3 Crypto Groups, which contain 3 Traffic Encryption Keys respectively, which can be used by any TETRA Radio to encrypt their traffic; however, there is a need to assign those TEKs to specific TETRA Radios in order to have an agreement of who should use which keys.

In the center, we can see UG 40, which is mapped to ISSI ranges 1000001 to 1000100 and 1000201 to 1000300; this User Group is also mapped to CG 1 and 2. With this mapping, we have now assigned the mentioned ISSI ranges to the TEK IDs from CGs 1 and 2.

The same happens with UG 45, in one side, it is mapped to the ISSI range 1000101 to 1000200, and on the other, it's mapped to CG 3, so the TETRA Radios in the range 1000101 to 1000200 can use TEKs with ID 7010, 7020 and 7030.

This kind of mapping mechanism is useful because we can divide the ISSI ranges in different chunks and assign them separately to different UGs, like in our example, were the ISSI range assigned to UG 45 is actually an intermediate range between the ranges assigned to UG 40.
Also, by dividing the TEKs into CGs, we can make sure that TRs receive some TEKs to be used only internally among a specified range (which can be for example an organization, UG 40 belonging to the Police, and UG 45 to the Fire department), and other TEKs are shared with other UGs (which can be used for example if there is a need for cooperation among departments).

3.3.2.3. Key Distribution

“If A and B each have an encrypted connection to a third party C, C could deliver a key on the encrypted links to A and B.” (Stallings and Brown, 2008, p. 618).

That is the very principle of this project; the KMF was created to be the trusted source of all the mobile (and non-mobile) devices in the network, the KMF is the heart of the encrypted connections, it is in charge to maintain all the devices used with E2EE operational in the network by providing them with Crypto Groups, which contain the necessary keys to establish communication with the rest of the allowed devices, all of this is done through an encrypted connection between the KMF and the recipients of the keys. The system uses different keys for different purposes, which are explained in section 4.3 of this thesis.

The keys for end to end encryption, as specified by the TETRA standard, are delivered via Short Data Service (SDS) messages, which are used as a bearer for the OTAK protocol in all TETRA system solutions.

Signaling for key distribution is explained in more detail in section 4.5.

3.3.2.4. Air Interface & End-to-End Encryption

In essence, Air Interface Encryption is the encryption performed between the TBS and the TR, it does not comprise any other parts of the network, that is, the voice/data travels in clear throughout all the wired network between the sending and the receiving TBSs, for this reason, E2EE needs to be implemented.
End-to-End Encryption, on the contrary of AI Encryption, encrypts the voice/data since it leaves the sending RT and it travels encrypted all the way to the destination, only then, the voice/data is decrypted for reception.

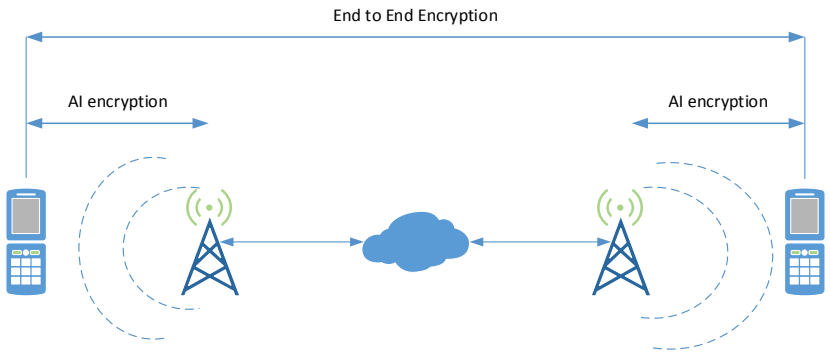

Figure 3 - Difference between air interface and end-to-end encryption

As explained in Stallings and Brown (2008), The E2EE cannot be used alone, due to the fact that if all the data is encrypted, the intermediate switches are not able to see the destination address, for that reason, in E2EE, only the payload is encrypted, and the header is left in clear, to overcome this weak point, Air Interface Encryption is used, encrypting the whole packet of data (header and the already encrypted payload) to be transmitted over the air interface.

4. The End-to-End Encryption Project

4.1. Quality criteria

The purpose of this project, as specified in section 2, is to deliver a mature solution with an acceptable quality, taking care of not affecting other parts of the system while introducing the new features related to E2EE and making use of them efficiently.

This is the first version of E2EE developed by SLC making use of OTAK delivery of keys for large-scale networks, for that reason, and because of budget and time constraints, it is not expected to deliver a full solution with all features included, instead, some management features have been left out, and the efforts were focused mostly on operational characteristics, having in mind that most major aspects of operations must be delivered with high quality, and other minor problems can be left behind if they do not affect operations, these problems will be addressed in subsequent versions of this solution.

4.2. Algorithms used

The project is planned to implement 2 encryption algorithms in the customer premises. First, the algorithm Advanced Encryption Standard with a key bitlength of 256 will be implemented as a transition from the current network security state to the E2EE with OTAK delivery capability, once that algorithm is in use and working as desired, an algorithm designed specifically for the customer, and called for that purpose Customer Algorithm (CAlg), will be implemented in order to customize the security for that specific customer.

4.2.1. Advanced Encryption Standard (AES256)

According to Stallings and Brown (2008), the AES is a symmetric block cipher algorithm with a block length of 128 bits and support for keys up to 256 bits. As mentioned in their book, Stallings and Brown state that a symmetric block cipher algorithm produces an output block cipher of the same size of the block of plaintext taken as input, and all the input blocks of text have a fixed length.

Operation of the Algorithm

The operation of the AES in simple terms, as explained in Stallings and Brown (2008) is as follows:

a) The plaintext is viewed as a matrix of bytes, this bytes are taken 4 at a time and placed into an array called “State”, this array, containing a block of 4 words of 32 bits (for a block of total 128 bits) is then modified by several transformations.
b) The first transformation is the substitution, for this, a table called S-box is used, for simplicity, this table is not shown in this document, however, we should see it as a simple table which maps every hexadecimal value to a new value (which is constant for all substitutions).

c) The second transformation is row shifting, a simple permutation is performed row by row. Since we are working with a matrix, each row contains 4 words of 8 bits, which is 32 bits per row, the amount of spaces that a byte is shift to the left depends on the position of the byte in the matrix, a detailed explanation of this operation is out of the scope of this thesis.
d) The next transformation uses equations over finite fields and its purpose is to mix the columns, for this operation each column operates individually, and each value of each column is mapped to a new value that is a function of all bytes in the column; for simplicity, the exact operation will not be described here.
e) The last and most important transformation is the round where we add our secret key. In this round we simply perform an XOR operation with the current block of data and a portion of our key.

All this process is repeated 9 rounds (9 more times) and finishes with a 10^{th} round of 3 stages (instead of 4 stages).
Figure 4 shows the stages of one round in a visual way:

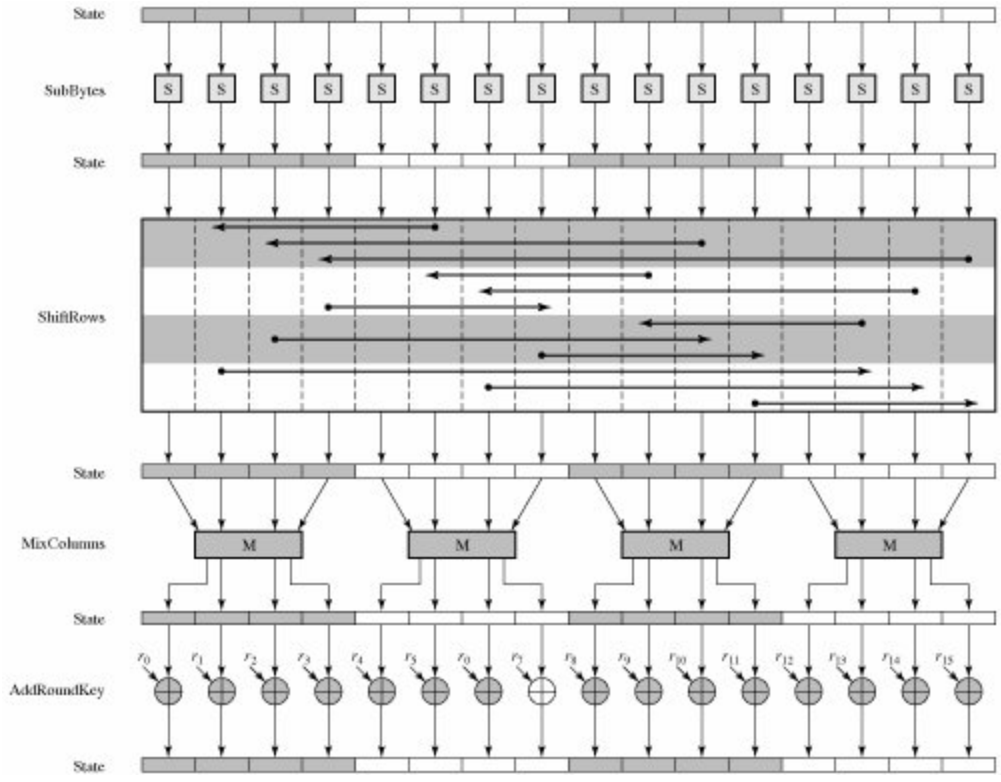

Figure 4 - Operation of AES256. Taken from Stallings and Brown (2008)

4.2.2. Customer algorithm (Calg)

As part of the E2EE solution, a customized encryption algorithm can be implemented for every customer; this algorithm can be used instead of the AES256 algorithm generally used for encrypting system's communications.
For security reasons, the customized algorithm for the customer for which this solution was designed is not known to anybody outside the customer's organization.
For the first part of this project, the customized algorithm will not be implemented; first, the solution will be implemented with the AES256 algorithm, after a period of time, when the solution is already in full operations, then the customer algorithm will be deployed in order to replace the generic AES256 algorithm.
It is important to mention though, that a customer can have its own algorithm, if he wishes so.

4.3. Encryption keys used in this solution

In this section, I will explain some of the encryption keys used in this project.

Due to the fact that for this thesis purposes I am participating mostly in the part involving the Key Management Facility, which is the new feature in this encryption system, I will describe only the keys that interact directly with the Key Management Facility and leave out the rest of the keys used in the system (for example, keys used for the air interface encryption by base stations).

4.3.1. Keys for communication (Traffic Encryption Key)

According to Finland's Airbus Defence and Space (2015, a, page 8), the keys used for encrypting audio and SDS messages in their systems are called Traffic Encryption Keys (TEK), these keys are used for encryption from one end to another, allowing to pass the encrypted data through all the system and only be seen in clear mode at the origin and destination.

In the customer's network will be a separation on how the TEKs are generated and managed, this separation will be organization specific, for example, if in the country in question there are several departments using the TETRA system (fire, police, army, etc.), they might be divided in organizations in order to keep them isolated from one another, and for this same reason, the TEKs will be generated locally in the organizations in which they will be used, then, the organizations will have to pass the keys to the KMF in an Out of Band (OOB) fashion in order to have them in the KMF database so it can then distribute them over the air to the intended users.
The general concept of generation, management and distribution of TEKs looks like in figure 5:

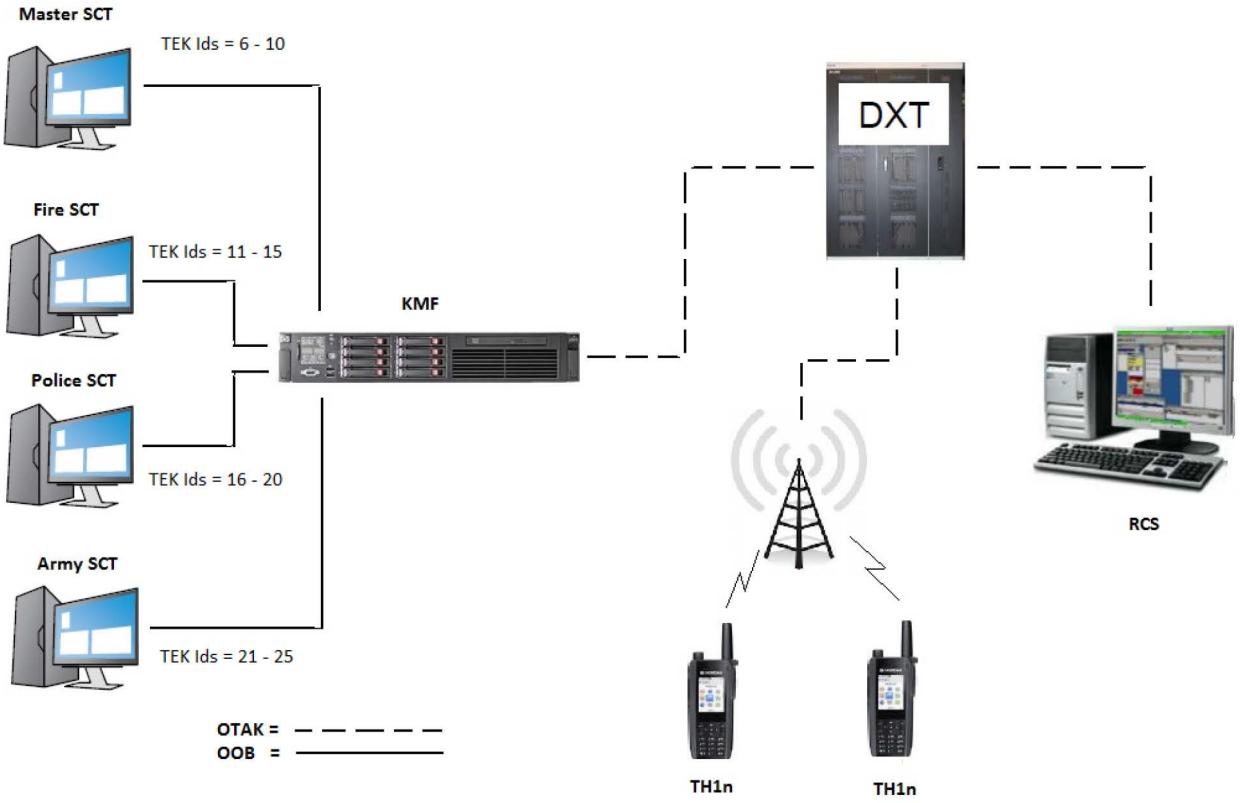

Figure 5 - Example of how the TEKs are distributed by organization

Every key domain in our solution consists of 4550 keys, and the key IDs are distributed accordingly through the domains, from smallest to biggest ID. For example, in domains from 0 to 9, the IDs contained in each domain will look like the following figure:

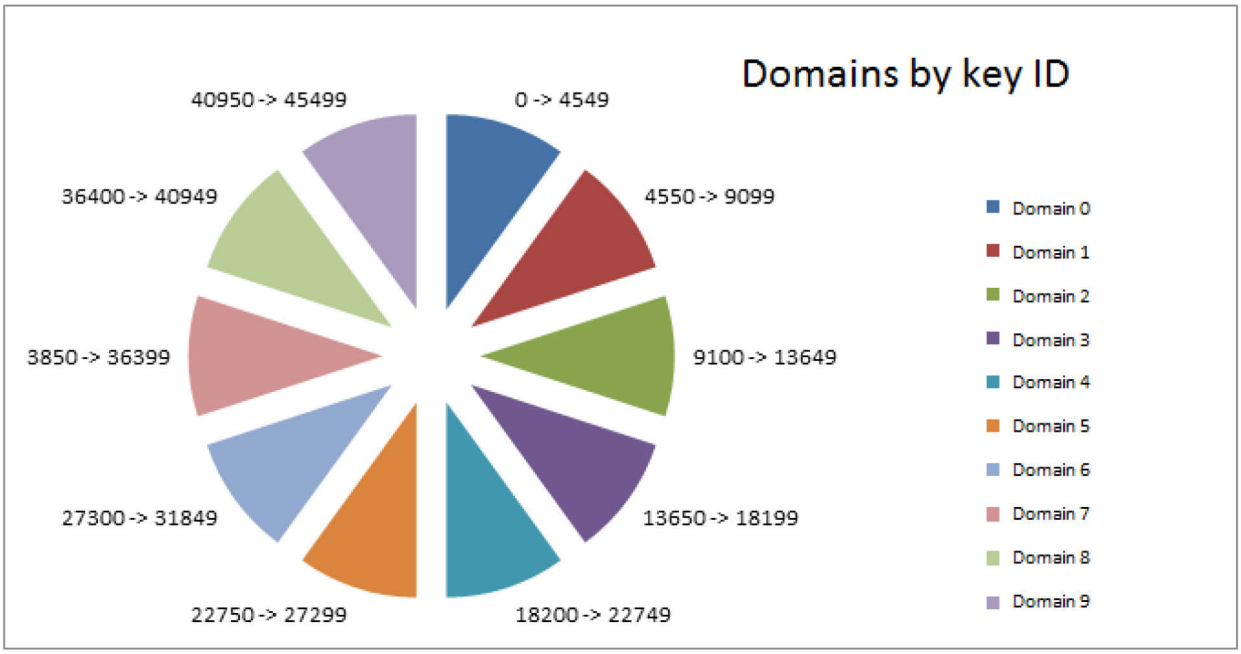

Figure 6 - Example of Key IDs in different domains

In the customer's case, the whole organization is divided in departments, and each department is assigned a set of domain IDs to use, hence, each department has its own unique set of traffic key IDs, in this manner, each department will produce and distribute its own keys to its users, and the key IDs will not overlap. The only case in which one (or several) domain IDs are shared among many departments is when referring to cross-department communication, for example, when using cooperation talk groups in case of a disaster situation (like a hurricane), the users from different departments need to communicate securely with each other, and consequently, they need to have common talk groups and the keys for encryption of communication in those talk groups.

One of the jobs of the Smart Card Tool is to generate keys, be it management keys or traffic encryption keys. As shown in figure 5, in our fictitious scenario, there are 4 SCTs, where 3 of them belong to specific departments (Fire, Police and Army), these SCTs produce encryption keys belonging to their own organization, which are used for intra-department communication. In addition to those SCTs, there is a 4th SCT with the name of Master SCT, the role of this SCT is to produce keys which will be delivered to the users of all departments in order to enable inter-department communication. The role of the Master SCT will be explained better in section 4.4.4.2.1 of this thesis.

4.3.2. Keys for Tetra Radios (KEK, SEK)

Besides of the Traffic Encryption Keys, there are 2 more keys used in the TETRA radios, these keys, called Key Encryption Key (KEK) and Signaling Encryption Key (SEK), are only used for management purposes, They are not meant to be sent over the air, and they do not encrypt any voice or data communication.

As explained by the Finnish branch of Airbus Defence and Space (2015), The KEK and SEK are used to encrypt the TEKs, as shown in figure 7:

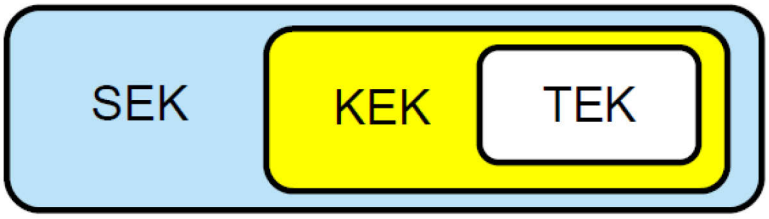

Figure 7 - Traffic Encryption Key sealed by management keys, taken from Finnish Airbus Defence and Space (2015)

As we can see in figure 7, when delivering encryption keys over the air, the keys meant to be used for encryption of voice and data (TEK) are first sealed/encrypted in a management key called Key Encryption Key, that has the function to encrypt keys (as its name says), in our case it encrypts TEKs. This whole package formed by a TEK and a KEK is in turn sealed/encrypted by another management key which is called Signaling Encryption Key, as its name implies, SEK is meant to be exposed on the signaling interface of the system.
Once this package is formed as explained here above, it is then sent over the Switching and Management Infrastructure (SwMI) and finally over the air to the TRs in the form of an SDS message.

At the TR side, the KEK and SEK must be programmed beforehand into the Smart Card (SC) in order for the terminal to be able to use those keys to decrypt the OTAK messages and recover the TEK(s) contained in them; this means that out of band operations must be performed to deliver the management keys (KEK/SEK) manually to both the TRs and the KMF.

The Smart Cards currently used by the Radio Terminals in our customer's network do not support OTAK messaging, and all Subscribers' SCs must be changed in order to introduce the firmware supporting OTAK, for this reason, it is convenient to initialize and load the new SCs with their corresponding KEK/SEK at the same time. The migration plan considers this issue and it is currently planned to massively program the SCs before they are delivered to the users.

Public key infrastructure does not exist in TETRA networks because TETRA networks are designed to be as secured as possible, and most internet infrastructure (such as public key servers) are not used; for this reason, our system is a symmetrical encryption system and needs to use out of band operations to deliver the management keys used later to encrypt the TEKs; this is because of the customer's wish to not use public methods.

4.3.3. Keys for KMF (FEK, BEK, IEK)

Besides of the previously mentioned keys, there are other encryption keys used in this solution which are not meant to be used for OTAK purposes, but instead for Out Of Band (OOB) purposes, these such keys, called File Encryption Key, Backup Encryption Key and Internal Encryption Key, are used mostly inside of the KMF, with the exception of FEK, which is used in all non-end user equipment (SCT, KSCC, KMF) when TEKs are transported OOB from one place to another.

FEK

The File Encryption key, which is the most important of all these 3 keys, is used for encrypting everything that needs to be moved from one computer to another.
The most common use for this key is to encrypt TEKs for transport between the SCT and the KMF; for example, in the case where we have a master SCT which produces TEKs to be used by different organizations, and thus, by different KMFs.

Figure 8 shows a Master SCT which is in charge of generating keys for 2 KMFs, each one corresponds to different departments; in our example, one KMF serves the network used by the police and the other serves the network used by the Fire department. These 2 networks usually would work isolated from each other; however, in case of a critical situation, such as a major accident or a natural disaster, both departments would need to communicate with each other. The KMFs used in each network would usually generate the keys for their own talk groups, but the keys for the shared talk groups between those networks cannot be generated by any one of those KMFs because they need to be shared. For security reasons, a security officer would generate the keys of the shared groups and distribute them OOB to the KMFs, in this process, the FEK would be used to seal the files transported OOB from the Maser SCT to the KMFs.

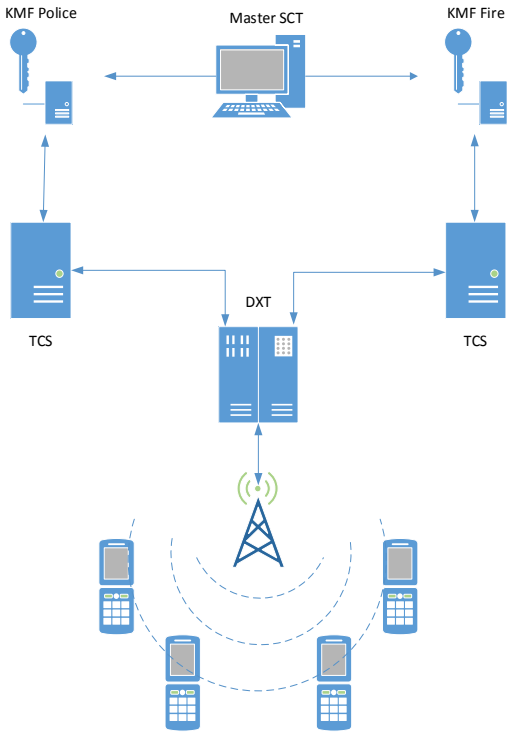

Figure 8 - Master SCT generating keys for several KMFs

BEK

Each KMF runs its own database, and every night, a backup of that database is performed, for security reasons, that backup is encrypted using the Backup Encryption Key. This key is used solely for this purpose and has no other use.

IEK

According to Finnish Airbus Defence and Space (2015, a), the Internal Encryption Key is only used to encrypt communication inside of a given KMF when the data travels from one component to another inside of the same server.

All these keys, FEK, BEK and IEK, must be the same in all KMFs belonging to one specific customer. In a disaster recovery situation, as for example, when there is a fire in one server's site but we have our data storage in a second servers' site, we can take a backup which has been stored in the second servers' site and use it to recover the information in a new server.

4.4. Types of devices used in the project

In this section, I will explain the most important elements of the E2EE solution.

4.4.1. Key Management Facility (KMF)

It is the most important element in the solution offered by this project and the element/area in which I am focusing the most for his thesis.

This element is a server which manages the automatic distribution of Traffic Encryption Keys over the air to all the radios in the network, as well as the distribution to other non-mobile equipment such as Radio Console Systems (dispatcher workstations used by operators in the main office to communicate with field agents) and Recorder Gateways (Recorders are devices used to record all communication in the network).

Besides of the previous mentioned task, the KMF also coordinates the usage of the TEKs in the storage of all the E2EE-capable equipment in the network. In TETRA networks, E2EE uses 3 keys, Past, Active and Future; this setting has the purpose of continuously use encrypted communications even when the expiration of old keys happen and new keys get into use.
For example, if a validity of a key expires in a Crypto Group, some radios will change the key to a new one before others; if only one key is in the internal storage at a given time, the communication will be lost for the time between one party and another in a communication channel change keys. On the other hand, if the currently active key stays stored in the internal memory of the device, but it is just marked as past, then receiving a call from another radio which is encrypted with that key (because the other radio hasn't yet changed to the new key) will end up in the end-user equipment being able to use the "past" key to successfully decrypt the call and communicate.

KMF contributes to this process by sending an activation message of the “future” key over the air to all E2EE-capable TETRA Radios, following by a “key load” message.

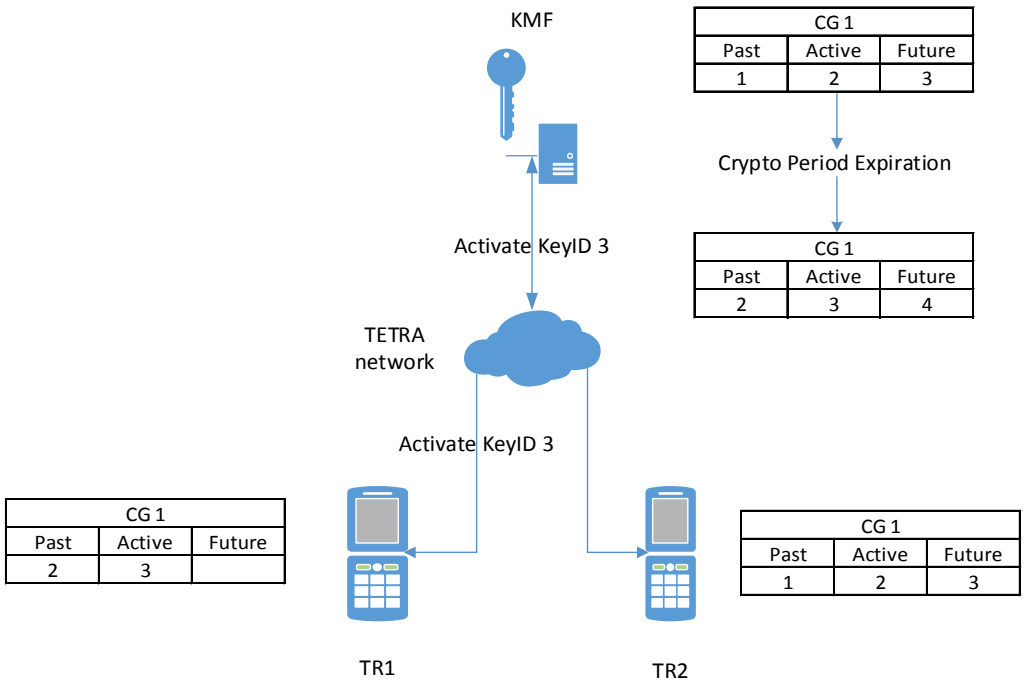

Figure 9 - Activation of the future key in CG1

In figure 9, I have illustrated a basic scenario where 2 TRs have different keys as their “Active” key for encrypting voice calls.
Let us assume that a given organization has a talk group which uses the TEKs in CG 1, and CG 1 is managed by that organization’s KMF; in the figure, for simplicity I have illustrated only the concerning elements. Let us assume as well that CG 1 has its three keys as follows:

- The key ID number or the Past key is 1
- The key ID number or the Active key is 2
- The key ID number or the Future key is 3

As shown in figure 9, as the time passes the period of validity of that active key comes to an end, consequently KMF automatically discards the Past key for that CG and moves the Active to the Past state; the Future key passes to the Active state at the same time that KMF generates a new key and places it in the Future state. Once this is done internally, KMF starts delivering keys to all the subscribers. In a big network where a KMF serves thousands of subscribers, not all radios would receive the Activation message and the new Key Load message at the same time as the other radios, this is mainly due to AI capacity constraints and the availability of the Main Control Channel of the TETRA air interface; because of this, one radio might receive an activation message many hours (or even days) before another; in the figure, we can see that one radio has already changed the active key to the one with ID 3, while the other has still the same old configuration, we can also see that the cell for the Future key is empty, that is because a KEY LOAD message needs to be sent to the radio once the KMF

produces the new key. In the configuration shown in the figure, we can see the importance of having a Past and a Future key, if TR2 tries to call TR1 and encrypts the communication with key ID 2, TR1 can decrypt it because the key is still in memory; however, if we would have only one key, the communication would not be possible.

4.4.1.1. Multi Generic Encryption Module (MGEM)

For encryption/decryption purposes, one of the most important parts in the KMF is the Multi Generic Encryption Module (MGEM), this is a board in the server which is in charge of encrypting/decrypting all outgoing/incoming communication.

Because in KMF there is no in/out audio, the only communication passing through the MGEM is the SDS messages for OTAK purposes. For every OTAK message arriving/leaving KMF, the MGEM encrypts/decrypts the TEKs and other data as explained in section "4.3.2. Keys for Tetra Radios".
All communication exchanged between the KMF and a given subscriber is always encrypted with the management keys (KEK/SEK), it does not matter if it's an association, a key load, an activation message or any other type of instruction.

The MGEM is also used in Radio Console Systems to encrypt/decrypt communication, in this kind of equipment, the MGEM is paired with another computer board called xGear16, which transmits audio streams; however, because the encryption functionality for RCSs was not tested in our Helsinki's laboratory, it is out of the scope of this thesis and I will not go into detail.

The MGEM looks like in figure 10 below:

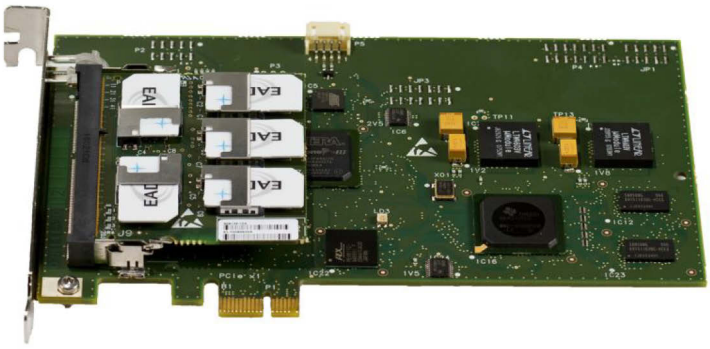

Figure 10 - MGEM, taken from the Finnish Airbus Defence and Space (2015, b)

4.4.1.1.1. Smart Card Module and Smart Card for KMF

It is possible to see in figure 10 a small daughter board inserted into the MGEM which contains 5 smart cards; this is called the Smart Card Module (SCM), and it is used in every MGEM in order to hold the SCs which contain the appropriate keys for encryption/decryption of communication.

In the context of the KMF, the SCM contains the Smart Cards for the Key Management Facility (or SC4KMF for short), these types of smart cards are different mainly in capacity from the SCs used in Radio Terminals. The KMF itself is designed to support an approximate amount of 40 000 users, in order to exchange information with that amount of users, the SC4KMF must have a bigger capacity than the ones used in TRs, each SC4KMF has the capacity to hold over 2000 pairs of management keys (KEK/SEK). In total, the version 1 of the KMF can hold 20 SCs, distributed in 2 MGEMs where each MGEM can hold 1 SCM and 1 SCM can hold up to 10 SCs (5 per side).

4.4.1.1.2. Authentication key for MGEM

The hardware of the MGEM and SCs used in radio dispatchers and the KMF is in essence the same, but the software is different; the MGEM and SC4KMF use a special authentication key to operate together, this key depends on the global location of the customer and has specific details for each customer that cannot be used in other customer's premises. The SC4KMF also has a different firmware loaded depending of the geographic region and the customer to which they belong, in this way, the SC4KMF cannot be reused.

4.4.1.2. KMF's Smart Card Configurator (KSCC)

The KMF's Smart Card Configurator is a tool developed for the sole purpose of initialize and program the smart cards for KMF. This tool cannot be used to program any other type of smart cards (as for instance, the SCs for TETRA radios).

It is a very simple tool which is easy to use; this tool will accept a SC4KMF and produce keys based in the algorithm contained in the firmware of the SC which is in our case the Advanced Encryption Standard with a 256 bit key. When generating the keys for SC4KMF, the KSCC will produce a set containing FEK, BEK and IEK; once these keys are produced, the tool gives us the option to save the keys to either the SC itself, the local PC or the security dongle (explained in the next section).
The keys generated must be saved to all the smart cards that will be used in the intended KMF in order for the smart cards to work when they are inserted in the MGEM of the KMF, the reason is because the FEK will be used in KMF in case TEKs are imported from a SCT (for example, the Master SCT) into the KMF, and the BEK and IEK will be used internally for backups and internal transfer of information.

4.4.1.2.1. KSCC's accessories – User Domain Tool (UDT) for KSCC and security dongles

The User Domain Tool (for KSCC, not to be confused with UDT for SCT) is another program which must be used before any operation in KSCC can be done. The sole purpose of UDT for KSCC is to create an authorized user which will be then allowed to log in and do operations in KSCC. The person in charge of using UDT and creating the authorized KSCC users must be someone completely unrelated to any activity in KSCC. This measure adds an extra layer of security to the overall process.
The process of how the whole mechanism works is as described in figure 11:

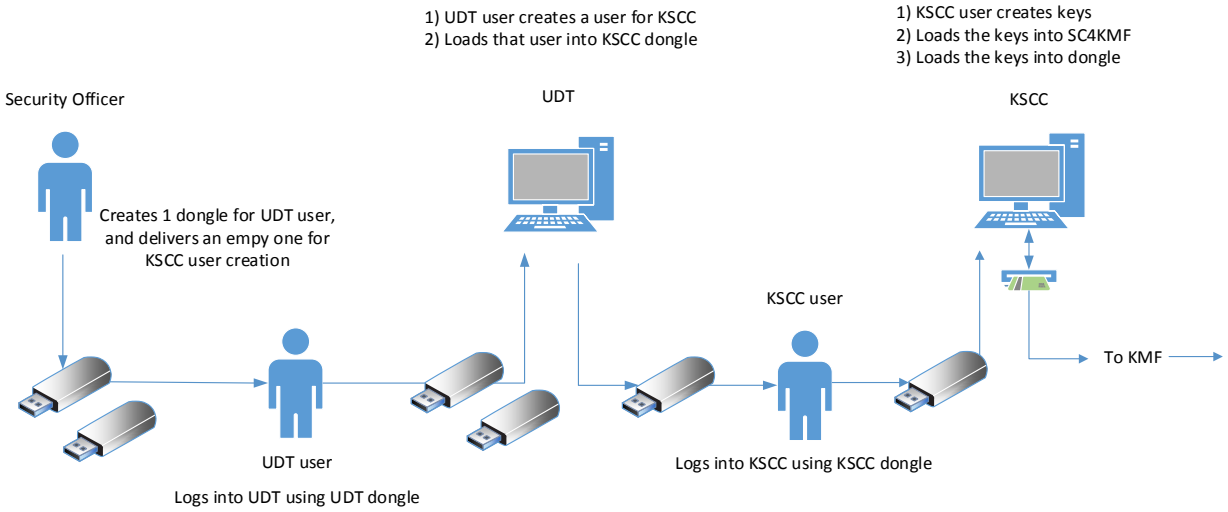

Figure 11 - Process to operate KSCC

We can see in figure 11 the process that needs to be done to operate KSCC and to configure the smart cards for KMF.
The operator must have a security dongle for using KSCC as well as for using the UDT for KSCC which serves as a type of electronic identification. Without the security dongle, the programs will throw a security exception and terminate.

The UDT operator must have a security dongle to log into his account in UDT; the company's security officer is in charge of issuing that dongle configured correctly. Once the UDT user gets this dongle from the security officer, he can log into his account in UDT and create a user for KSCC; after creating that user, it must load that information into another security dongle which will be delivered to the person in charge of operating KSCC.
The KSCC user can log into KSCC and create the necessary keys for configuring a SC4KMF (FEK/BEK/IEK) after receiving the KSCC security dongle, then load those keys into the smart cards.

Finally, if TEKs will be imported from SCT to KMF via out of band, as for example the cross organization group encryption keys, then KSCC user must load the FEK into its security dongle and transport it OOB to SCT in order to be able to seal the TEKs to be moved with FEK for secure transport. When KMF receives Out Of Band TEKs, it will be able to unseal the container because it already has the FEK loaded into the smart cards inserted in its Multi Generic Encryption Module.

4.4.2. TETRA Connectivity Server (TCS)

By Finnish Airbus Defence and Space (2015, c) “The TETRA Connectivity Server (TCS) provides a high-level interface for connecting customer-specific applications to the TETRA System securely and efficiently. This interface is called the TCS Application Programming Interface or TCS API.”

In short, we could say that the TCS is the door between SLC's TETRA Systems and the rest of the world when it comes to custom applications which need to do a specific job in SLC’s TETRA systems.
There are other interfaces to the TETRA systems, for example, voice is transferred by the TVG (TETRA Voice Gateway) as well as Inter System Interface (ISI) and Generic 4 Wire Interface (G4WI) for other purposes; however, when it comes to signaling between the TETRA network and third party applications, the TCS is the bridge which needs to be used.
For the purposes of this thesis, only the connection through the TCS is important and for that reason I will not explain the rest of the interfaces into the TETRA system.

The TCS API is a piece of software that can be run in a regular Windows 7 computer or alternatively in a Windows server; a Windows server is used for OTAK purposes, the reason for this is that TCS running in a Windows 7 computer is mostly intended to run together with a dispatcher application for which a single user TCS software is used. For OTAK purposes; however, the TCS can be a separate generic multi user server which can be used by the KMF and other third party applications at the same time.

4.4.3. TETRA Base Station (TBS)

In cellular communications, “the BTS provides the direct communication with the mobile phones.” Poole, I. (2006). According to this author, The base station is the link between the mobile equipment and the network infrastructure, being this Base Station comprised of the antennas themselves and the electronics in charge of multiplexing and amplifying the signals; on top of this, there is a base station controller which is in charge of managing the resources of the base station, and in charge of communicating with the switching center (DXT, explained in section 4.4.5 of this thesis).

In the case of SLC, all the elements mentioned above are together in the same physical device, the TBS contains the base station controller, radio transceivers, radio frequency multiplexers and power supplies; it is capable of passing signaling, voice and data traffic to the DXT, as well as to act as a standalone equipment routing the voice calls within the radios directly under its coverage.

For E2EE point of view, there is no special role for the TBS, its sole purpose is to pass the OTAK messages to and from the TETRA Radios.
The TBS has a core role in Air Interface encryption and authentication of Mobile Subscribers, however, that subject is not covered by this thesis, for that reason, I will not get into details.

4.4.4. TETRA Radios (TR)

These are mobile devices used for TETRA networks, they are not like regular commercial phones because the environment of operation is quite different and they need to be resistant to more extreme areas of operation compared with regular consumer phones, certain TETRA Radios support End to End Encryption using OTAK for delivery of keys, we used a model called TH1n radio.

4.4.4.1. Smart Cards for TRs

As mentioned in section 4.4.1.1.1, there are 2 different types of Smart Cards, those meant to be used in the KMF's MGEM and the rest which can be used in TRs and RCSs. The latest have a firmware designed to hold only 1 pair of management keys and the rest can be filled with Crypto Groups and TEKs.
The SCs for TETRA Radios are also designed to be controlled remotely in the sense that their keys can be erased over the air in case the radios are lost or stolen.

The Smart Cards for TETRA Radios have to be configured only once manually at initialization in order to write into them the Individual Short Subscriber Identity (ISSI) address of the KMF and their own pair of management keys (KEK/SEK); with this information, the SC is able to contact the KMF when the TR is switched on in order to request TEKs.

4.4.4.2. Smart Card Tool (SCT) for TRs

Smart Card Tool is a software used to configure subscriber SCs both from TETRA Radios and from Radio Console Systems. Besides of configuring SCs, the SCT can also create keys to be used as Traffic Encryption Keys; for example, if keys need to be distributed by several KMFs such as in the case of the cross-organization Crypto Groups.

The following image is an example of the SCT:

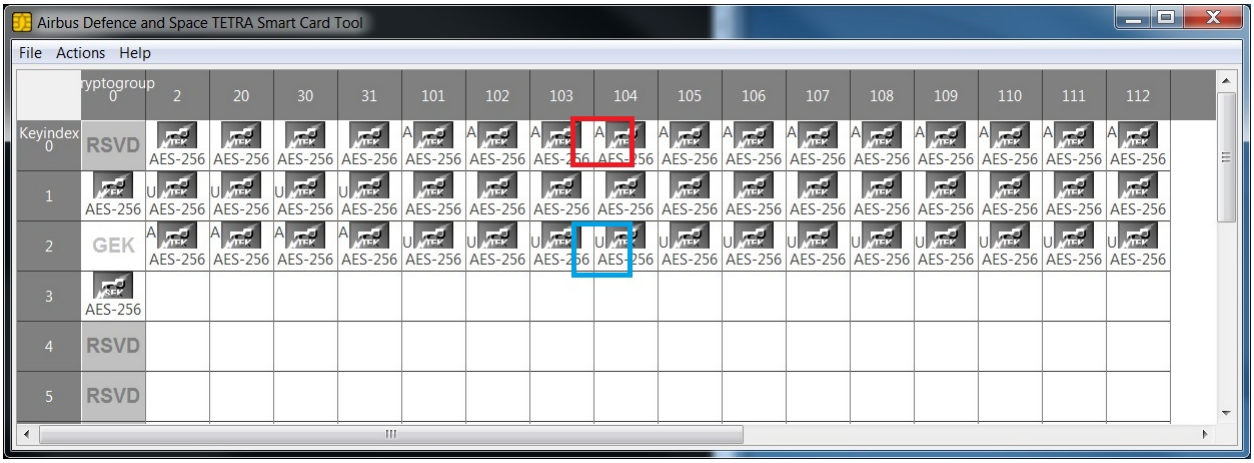

Figure 12 - Example of Smart Card Tool

In Figure 12 we can see how the KeyStore of a Smart Card looks like; in this specific SC we have 16 CGs which have each 3 TEKs, for each group of 3 keys there is one in “Past” state, marked with a “U” (short for Used), another in “Active” state marked with an “A”, and an unmarked key which is in the “Future” state and hasn’t been used yet.
In the left part we also see 2 keys in the Crypto Group 0 column and although this is not shown, those keys are the Key Encryption Key and the Signaling Encryption Key. In this example the corresponding slots have been already filled; however, because CG0 is not used for traffic keys, the slots in its column are marked according to the type of key it holds, at the moment of initializing the SC is possible to see KEK or SEK engraved in them.

4.4.4.2.1. Master and Organizational SCTs

In Figure 13, we can see that there is a so called Master Smart Card Tool together with a couple of departmental SCTs which supply Traffic Encryption Keys from different domains to the 2 KMFs in order to be sent over the air.
The role of this Master SCT is to provide the same set of keys to several KMFs in order to be distributed to all the users contained in the organizations which those KMFs serve.

In Figure 13, we can see that one KMF is connected to its departmental SCT and to the Master SCT; for example, in the case of the Fire department, it has a KMF designated in the illustration as “Fire KMF” which is in turn connected to a Fire SCT and at the same time to the Master SCT.
Let us assume that the Fire department has an internal talk group with number 6002563, and that this talk group is associated to CG1, at the same time a cross-organizational talk group with number 2008541 exists and this cross-organizational talk group is assigned to CG2.
In our example, Fire SCT will generate the keys for CG1 and deliver them (OOB) to the Fire KMF. At the same time, Master SCT will generate the keys for CG2 and deliver them (OOB) not only to the Fire KMF but also to the Army KMF; in this way, the users of the Fire department will be able to communicate with each other through talk group 6002563 in a group call encrypted with the keys contained in CG1

which are delivered from the Fire KMF to the users of the Fire department. But also, when necessary, the users of the Fire department will be able to communicate with the users of the Army department through the cross-organization talk group 2008541 in a group call encrypted with the keys contained in CG2 which were delivered to the users of both organizations by their respective KMF.

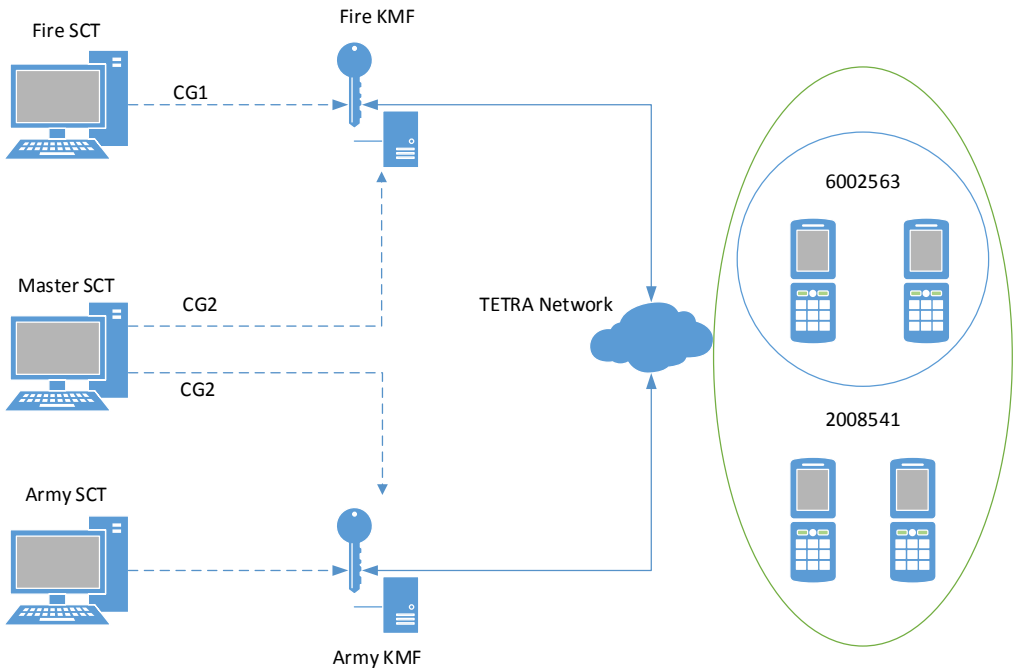

Figure 13 - Master SCT delivers keys to several KMFs

In our previous example we are assuming that the KMFs rely in a departmental SCT to generate the keys; I explained it like that in order to make the role of the Master SCT clearer. However, in reality, the TEKs for intra-agency organization are generated locally in their respective KMFs for convenience and the Master SCT generates only shared TEKs.

4.4.4.2.2. SCT's accessories (UDT and security dongles)

The SCT also makes use of a User Domain Tool in a similar manner as the KSCC, this serves as an extra security measure because it is operated by another person whose role is to only create users and assign those users a key domain for use in the SCT.

The concept of the SCT´s dongles and UDT is in principle the same as in KSCC, with some few differences:

a) The KSCC is a software derived from the SCT, thus, the accompanying elements and process of operation are very similar; however, being the parent software, the SCT software has a richer design.
b) SCT uses different domain ranges for TEK creation, on the contrary of the KSCC, which uses only 1 domain range for FEK/IEK/BEK creation.

Figure 14 illustrates the process for using the SCT; the process is very similar as that of the Figure 11 corresponding to KSCC, except for the steps performed in UDT and SCT stages, which are different:

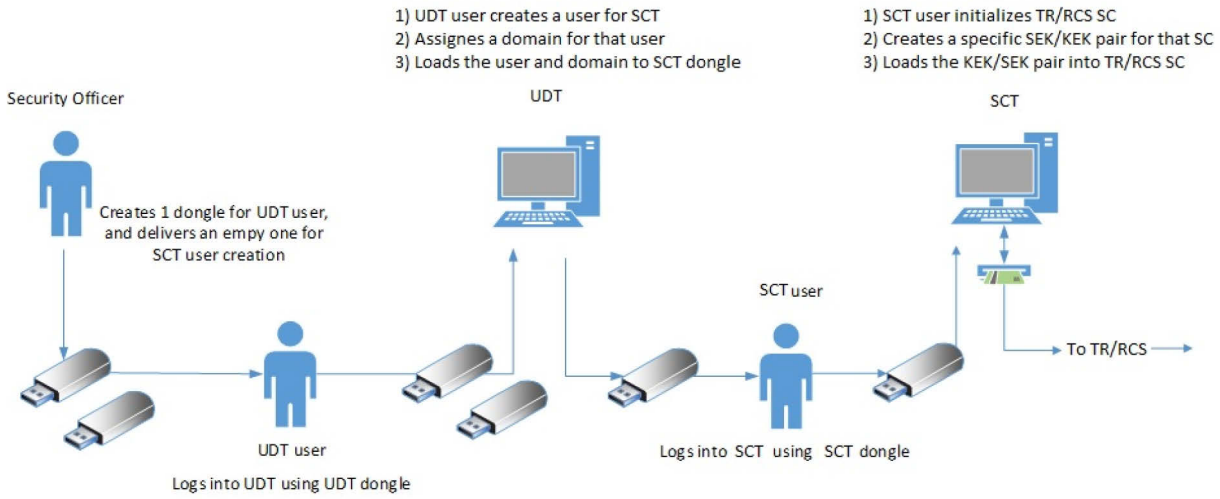

Figure 14 - Process for SCT user creation and usage

It is imperative to make clear that the UDT used for SCT and for KSCC are not the same, each tool (SCT and KSCC) has its own version of User Domain Tool, and the users created for example in the UDT from SCT, will not work in KSCC and vice versa.

4.4.4.2.3. Encryption certificates

For an extra layer of security in OOB management of keys, the E2EE solution uses SMIME certificates on top of the FEK encrypted files.

Every time a file, whatever it is (TEKs, Associations, KEKs/SEKs, etc.), is moved from one computer to another via an external memory storage, the file is automatically sealed with an SMIME certificate previously installed in the computer where the file is being produced. The same certificate must be installed in the destination computer in order to decrypt the file.

SMIME is not used when sending OTAK messages, it is merely used for OOB management purposes.

4.4.5. Digital eXchange for TETRA (DXT)

In cellular systems, the Mobile Switching Center is the most important component of the whole network; in SLC's TETRA implementation, that device is called Digital eXchange for TETRA (DXT) and its main purpose is to route the voice calls and data between all the subscribers in the network.

There have been many DXT models throughout the history. More than 20 years ago the DXT was a huge switching cabinet that could easily take a full small room; at the time of writing this thesis, however, the DXT is a box smaller than a dish washer machine and uses the Advanced Telecommunications Computing Architecture (ATCA); most of its components are virtual computer units doing the functions of what once were enormous physical devices accommodated in a continuous array of cabinets.

Figure 15 below shows a picture of the DXT:

Figure 15 - DXTA by SLC, taken from Sturtzel, A. (2015)

The DXT has many different interfaces and it's capable of connecting to different devices in order to obtain their services for different purposes.
Some of the main tasks and functions of the DXT are enumerated below; the services from the third point onwards might or might not be in use depending on the customer:

- Maintain a numerical tree for traffic routing
- Switch calls and data transfer among different subscribers
- Manage trunk links with other DXTs
- Manage charging services
- Manage statistics (for traffic, load, etc.) and reporting
- Together with the Enhanced Packet Data Gateway (not covered in this thesis), routes packet data to its destination

- Communicate with external systems such as PSTN, PABX, ISDN and other cellular radio networks
- Connect different applications to the TETRA network such as voice dispatchers, management applications, network monitoring centers, etc.

Due to the extensive nature of this specific device, it would take a whole document to describe every function of it, for that reason and because the focus of this thesis is in security and encryption management, I will not get into more details regarding switching telephony

4.5. From key generation to call establishment, the process explained

In any kind of encryption mechanism is recommended to change the encryption keys every certain period, for that reason, in our End-to-End Encryption system the keys used for communication are changed in a regular basis configurable by the user. In the case of our tests in the lab, we used a 1 hour for the crypto period expiration; however, in the customer network it could be several weeks.
In this section, first I will explain shortly how the keys are generated, then, once the keys are ready, they need to be distributed, I explain the signaling concerning their distribution in the second section; and at the end when the keys are in the end user equipment, I explain how a call is established.

4.5.1 Key generation

Keys are generated in 2 ways in our End-to-End Encryption solution, directly inside of the Key Management Facility before they are distributed to all subscribers, or in an external software hosted in a different computer called Smart Card Tool.

When all the TETRA Radios that need to communicate are served by one single KMF, then that KMF can generate and distribute the keys by itself as seen in figure 16.a:

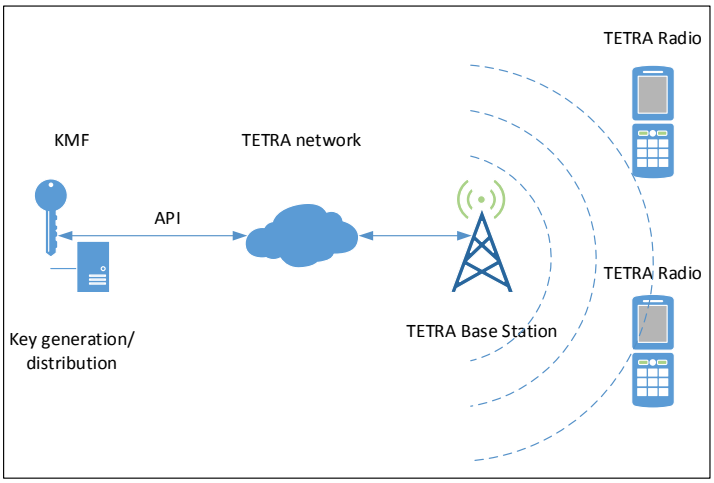

Figure 16.a - Keys generated inside of KMF

However, when the TETRA Radios which need to communicate are served by different KMFs as in figure 16.b, where KMF 1 serves TR 1, and KMF 2 serves TR 2, an external software needs to generate the keys and passed them to the corresponding KMFs for their distribution; otherwise, if one of the KMFs would generate them, the other would have no idea of what keys are being distributed

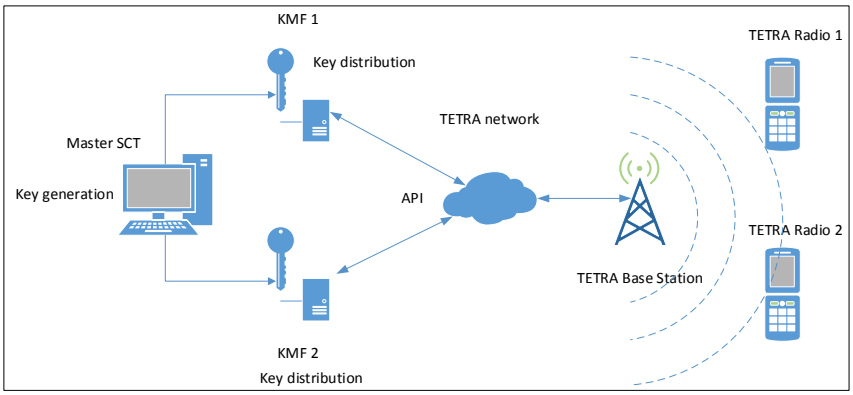

Figure 16.b - Keys generated outside of KMF

In our End-to-End Encryption solution, the party in charge of generating the keys hosts a physical board containing the key generation machine, this board, called Multi Generic Encryption Module (explained in section 4.4.1.1), contains a number of Smart Card, which in turn contain the algorithm necessary to generate the keys.

4.5.2. Key exchange

Several steps must be performed first for key generation and exchange in order to establish a successful end-to-end encrypted call using this solution. These steps, as well as other details in signaling is described here below:

1) As a pre-requisite, all the devices intended to work in end-to-end encrypted mode, must first receive a properly configured smart card (SC), these smart cards must be pre-programmed with a set of management keys (Key Encryption Key/Signaling Encryption Key) which are used to decrypt the keys used for communication (Traffic Encryption Keys) when these are received encrypted on messages through the Over The Air Keying (OTAK) protocol.
   During the pre-programming of a smart card also the Key Management Facility's (KMF) Individual Short Subscriber Identity (ISSI) must be included, in this way the TETRA Radios (TRs) will be able to contact the KMF during registration in order to receive the rest of the required information.
   Because there are thousands of mobile users in the customer's network, all the required smart cards will be first massively pre-programmed and then distributed to all users.
2) Once the SC is properly pre-programmed and inserted into a TR, it needs to be switched on and when this happens the TR registers to the TETRA network in a regular way, at the same time, the TR will send a registration message to the KMF to register as a client for OTAK protocol messages reception, making in this way itself available to receive OTAK messages containing TEKs which will be used later to encrypt/decrypt communications (voice calls and text

messages). This registration message is encrypted with the KEK previously pre-programmed in the SC.

3) The first time the KMF receives a registration message (first time an end-to-end encrypted TR is switched on), it will bond the ISSI address from which it was received together with the management key (KEK) in which the registration message was encrypted; in this way KMF knows who is using what key for signaling (exchange of OTAK protocol messages).
4) Once KMF registers a given TR into its database, it will look up the User Group (UG) to which the TR belongs and deliver all associations and encryption keys corresponding to that specific TETRA Radio's User Group.
5) When the TR in question has received all associations and encryption keys corresponding to its UG, it will be able to start communicating to other TRs.

Before I continue explaining the procedure to establish an end-to-end encrypted call, I want to present in a graphical manner the steps 1 to 5 previously described.

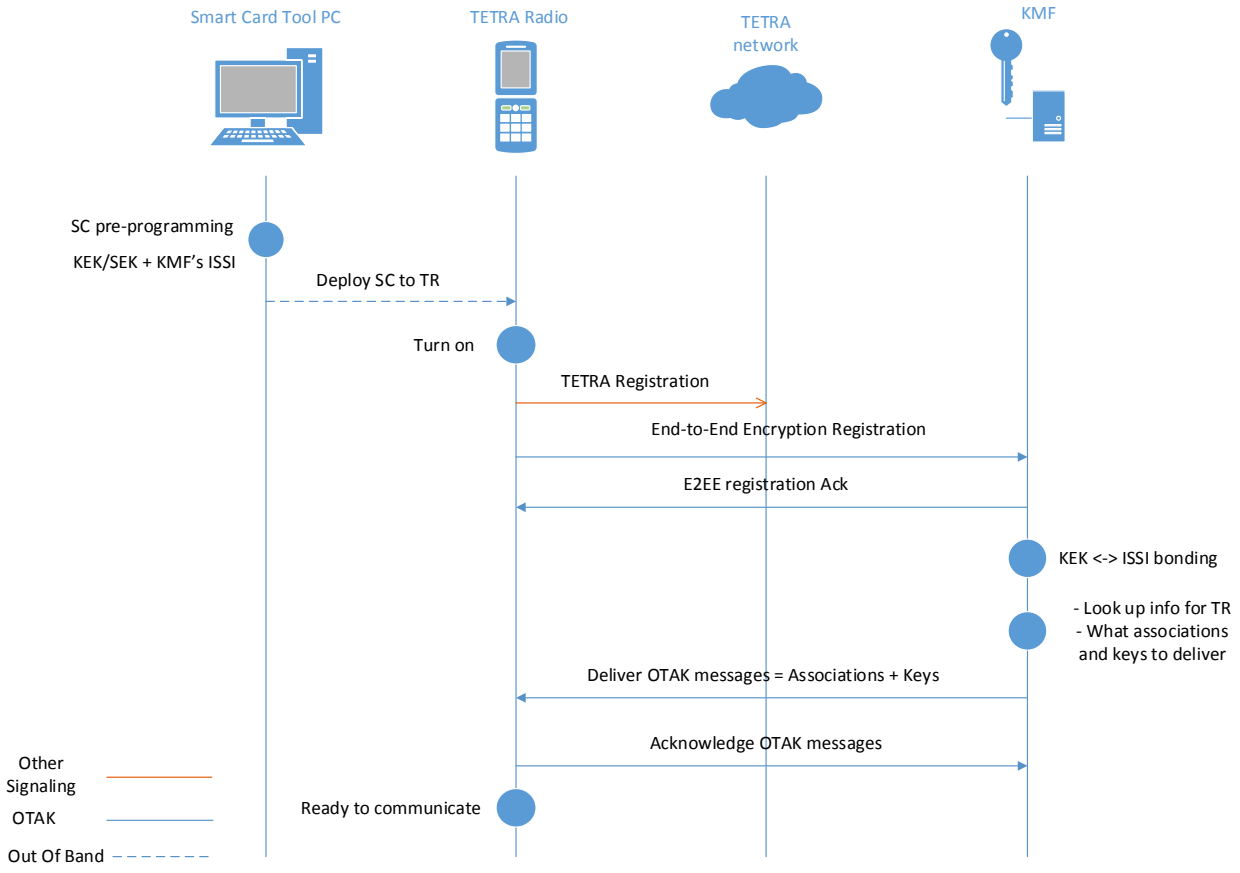

Figure 17 - Sequence diagram for key exchange signaling

4.5.3. Call establishment

Once all the signaling to deliver the encryption keys described in section 4.5.2 is done, the TR can start communicating. The procedure to establish a call using our End-to-End Encryption solution is as follows:

1) The user dials a number for an individual call or alternatively pushes the Push-To-Talk button on a previously selected Talk Group to start a call.
2) Let us concentrate in the case of an individual call for this example. Once the user presses the "Dial" button, the TR looks up in its SC for the list of associations previously delivered by KMF, when it finds a suitable match for the destination address, it looks for the "active" encryption key contained in the Crypto Group mapped to the destination address.
3) Having found the correct encryption key, the TR encrypts the voice call with that crypto key and starts sending the voice packets (circuit switched TDMA based) over the network.
4) The steps 1 to 3 assume that the corresponding key has been delivered to the other end of the voice call using the procedure in section 4.5.2, hence, the recipient is able to answer and decrypt the call.

Last but not least, here is a graphical representation of the procedure to establish a call using our End-to-End Encryption system:

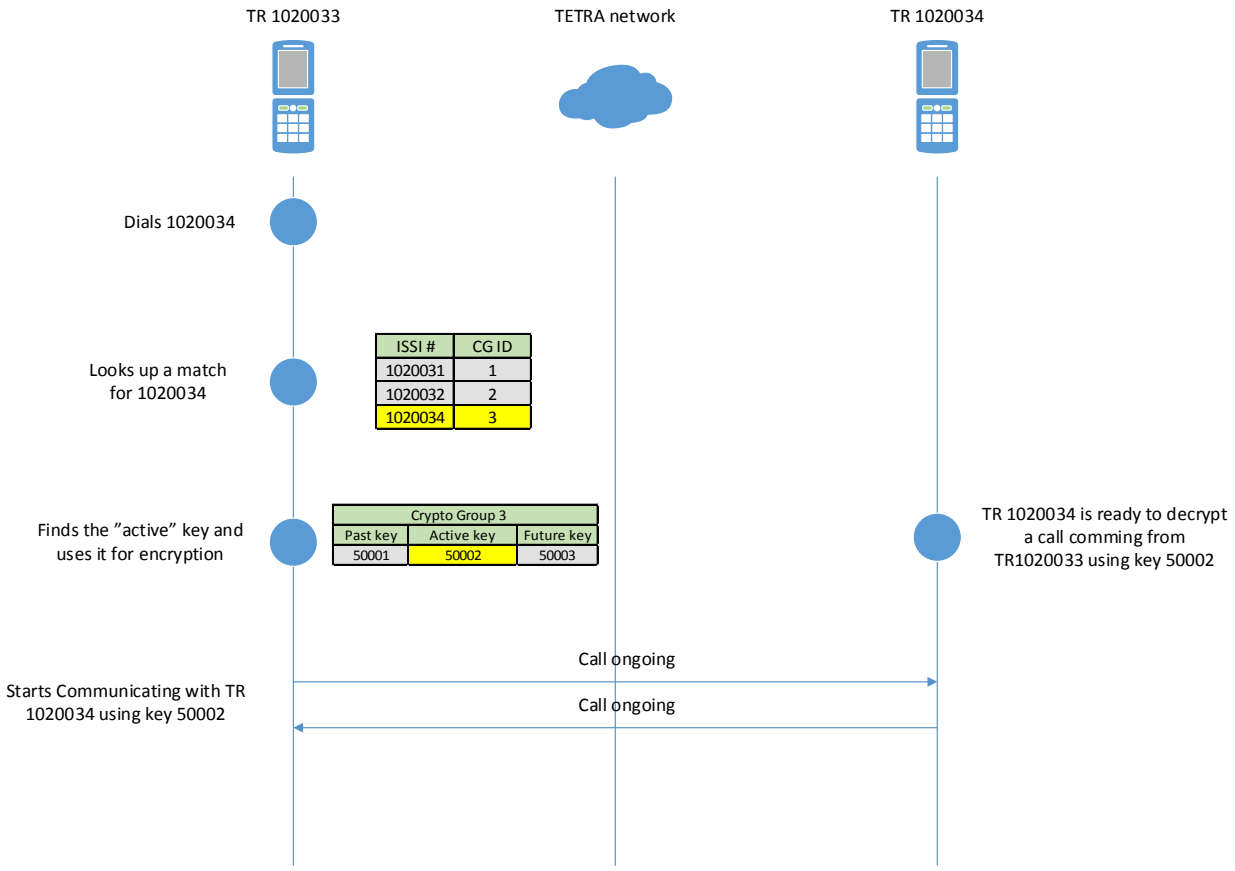

Figure 18 - Process of call establishment using our End-to-End Encryption system

4.6. The project explained in stages

In this section I will talk about the project stages from the point of view concerning the responsibilities of the Helsinki's office.

This was a multi-site project, in which teams from 3 different locations were involved (Paris, Helsinki and Jyväskylä), and the tools/processes were divided so each site had its own responsibilities in the overall success of the project. For this reason, I will describe the stages undertaken in Helsinki, in which I was directly involved.

Figure 19 illustrates an overview of the project's phases in Helsinki:

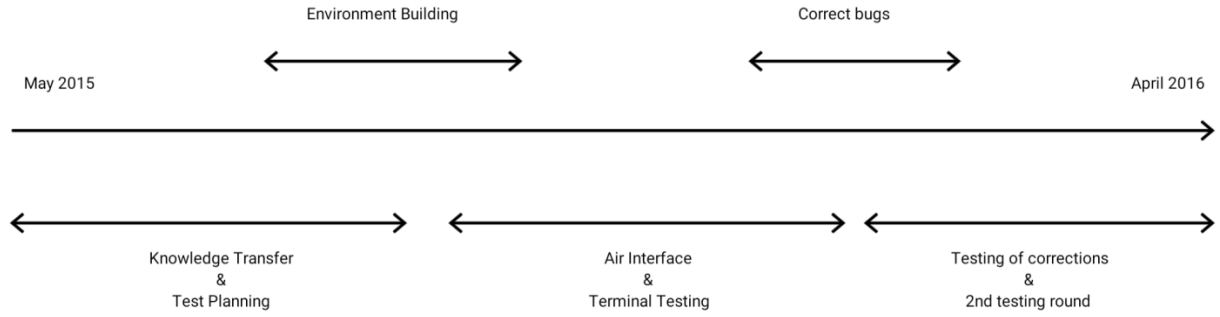

Figure 19 - Illustration of the phases of the project in Helsinki

4.6.1. Planning Phase

At the beginning of this phase, my main task was to absorb an overall knowledge of the design and underneath technology of both the solution in general and the customer's network architecture, in order to be able to do the real work later. For this purpose, I had several meetings with the network architects in order to gain knowledge of how the customer's TETRA network was structured, what was the expected traffic over which our traffic load profile should be based, as well as what was the crypto configuration expected to be present in the customer's network, so then with this information generate a verification strategy/test plan tailored for our customer's needs.

At the end of this phase, I focused on preparing a verification testing strategy definition and a test plan for KMF's verification testing over the air interface. I generated 2 files, an overall verification strategy and a test plan, explained in section 4.6.1 of this document.

4.6.2. Test environment

The environment building started during my training period in the KMF tool, at that time, my colleague in charge of the air interface signaling analysis started to deploy the infrastructure concerning the air interface link between the base stations and the radio terminals (for convenience and regulatory issues, we use cabling in the lab instead of transmitting traffic on microwaves over the air).
In Airbus Helsinki's lab, there is a specific area dedicated to testing the TETRA tools over the air interface, so he only needed to do small modifications to an already deployed environment, such as changing configurations in the TBS and adding more dock points for TRs (car kits to hold the TRs, computers to monitor their traffic and cabling to isolate the AI from unwanted external noise).

Once I received training concerning the KMF tool, and we had the necessary physical equipment (servers, smart cards, etc.) I started to build the part of the network concerning the KMF (Installing and commissioning the server, then integrate it to the TETRA network through the TCS).
The AI network architecture is specified in section 4.6.1.2 of this thesis.

4.6.3. Air Interface (AI) capacity testing

KMF product testing was performed in France. Once that testing was completed, and the KMF product team has released an official version of the KMF, we started to test the solution in Helsinki focused in the impact of it on the customer's mimicked TETRA network we had built in our lab.

This testing's primary purpose was to find how the network overall would be affected by this new solution taking into account the Air Interface capacity between the TETRA Base Stations and the Radio Terminals.

The Air Interface is usually a bottleneck in the traffic flow because of the physical characteristics of the radio spectrum, which will affect the performance of the rest of the network, for that reason, it is mandatory to test how a new solution's introduced traffic would impact the normal traffic such as voice calls, TR's tracking signaling, etc.

4.7. Thesis work in practice

This section provides the details of my contributions to the project.

The format in which I decided to present this section corresponds to the timeline of the events and the phases of the project.

I start by explaining the activities I performed during the planning phase, what documents I produced, and an overview of the information contained in those documents.

I then dig into the details of the Air Interface testing, I talk about the network architecture and configuration we deployed, how the test plan for AI was performed, as well as about the defects found and their corresponding solutions.

4.7.1. Activities during the planning phase

Since this was the first stage of the project for Helsinki's office, the very first activity I had was to absorb the knowledge concerning E2EE, for that I started by studying the available documentation because I was not familiar with E2EE in TETRA networks before.
It was not easy to learn the implementation of E2EE with OTAK messaging because there was not much formal documentation about it since this is the first version of the KMF and also a new implementation of the OTAK standards for E2EE defined by TETRA SFPG (Security and Fraud Prevention Group), and by consequence, our own documentation regarding the implementation of the standards in our solution was not yet written.
I was able to gain the mentioned knowledge by having meetings with network architects and reading internal documents they had written during the previous months when they designed the solution, as well as traveling to France (where the KMF had been developed and tested at a product level) for a week of hands-on training in installing and configuring KMF.

The second most important thing during the planning phase was to define a test strategy to be performed in Helsinki and a test plan for the testing activities assigned to us.

In the following 2 subsections I explained my contribution to this test plan and strategy.

4.7.1.1. Load profile and testing strategy definition

During the first months of my thesis, and after the initial phase of familiarizing myself with the project, I started to work in defining a strategy for testing the corresponding parts of the project assigned to our office in Helsinki. This strategy comprised the definition of a traffic profile to load the networks during our testing, the requirements to build a suitable test environment for each of the 2 testing phases, and an outline of the general strategy to be performed during the following months.

I will explain in this thesis the main points covered in the test strategy document in a shortly manner, but covering the most important information.

Testing strategy definition

In the E2EE project, I proposed and created the test strategy for the responsibilities assigned to our office in Helsinki. In order to do this, I created a document in which I specified the main points to cover during our tests as well as how we would achieve that testing.

The test strategy document describes what kind of tests we would do in Helsinki (OTAK traffic affecting the AI capacity), as well details of what is needed for those tests.

In that document I propose the network architecture to be used in our laboratory (explained in section 4.6.1.2 of this thesis), as well as the investments we needed to do to have the necessary equipment to perform those tests.

The mentioned document also describes the scope of our tests, which is, in a simple matter of saying, to find out how an artificial network build as a down-dimensioned copy of the real customer's network is affected when the signaling traffic generated by this solution is loaded into it. On top of this point, our tests are intended to verify the components of the OTAK E2EE solution at a system level, and report every found defect to the corresponding area (KMF, TR, or TETRA network side) for correction and retesting before the solution is deployed in the customer's network.
The load profile traffic (explained here below) is also specified in the test strategy definition, in order to explain the facts in which the proposed down-dimensioned network architecture was based.

Air interface load profile

The load profile contains the traffic to be expected to happen when the E2EE solution with OTAK key delivery is deployed into the customer network.
Because the main purpose of the testing is to check how the air interface will behave when the new OTAK messaging is introduced and then in use at the customer's network, the traffic in this profile is similar to that of the real conditions.

The load profile has to be divided into 2 parts representing the current traffic on the network, and the new OTAK traffic is then introduced.

The current traffic, which I called "Background traffic", is the combination of group calls, individual calls, data messages (SDS), etc. This mentioned traffic is happening before, during, and after the new OTAK messaging traffic is introduced.

The new traffic, which from now on I will call only "OTAK traffic", includes all the messaging necessary to register, receive and acknowledge all messages sent by the KMF to the tetra radios, being encryption keys, associations to use those encryption keys, activations of the keys and any other type of messages.

During the registration of the mobile subscriber to the network, there is extra signaling besides of the OTAK traffic due to the protocols proper of the TETRA standard, for simplicity, this extra traffic is specified separately.

All numbers described in this load profile are averages.

Background traffic

The following table describes the breakdown of the signaling in terms of unit per time, it is estimated that there are in average 65 tetra radios per base station at any given time, so the following amounts are based in this number:

Background traffic					
Protocol	Direction	Amount	Duration (s)	Repetition	Comments
Group call		325	20	Hour	
Late entry	Down	1		Second	As a broadcast
Individual call		65	180	Hour	
PSTN/PABX call		65	180	Hour	
Individual Status	Any	65		Hour	16 bits
Individual SDS	Any	65		Hour	64 characters. Acknowledged
Group Status	Any	65		Hour	16 bits
Group SDS	Any	65		Hour	64 characters. Acknowledged
Group Management SDS	Down	129		Work shift	
Roaming (Inter-DXT)		129		Hour	
Handover (Intra-DXT)		65		Hour	
Packet data		104,4643		Hour	milli-earlang
AVL	Up	129		Minute	

Table 1 - Background traffic for AI testing

Due to the nature of this testing phase, the traffic proposed is based in the amount of traffic generated per subscriber multiplied by the number of subscribers found in a single base station. In short, the amount of traffic per subscriber is as follows:

- 5 group calls of 20 seconds of duration per hour per subscriber
- 1 individual call of 3 minutes of duration per hour per subscriber
- 1 call to external systems (eg. PSTN) of 3 minutes of duration per hour per subscriber
- 1 individual status message of 16 bits per subscriber per hour
- 1 individual SDS message (and its acknowledgement) of 64 characters per subscriber per hour
- 1 status message of 16 bits to a group per subscriber per hour
- 1 SDS message (and its acknowledgement) of 64 characters to a group per subscriber per hour
- 2 subscribers moving to/from an area covered by another DXT (roaming) per hour
- 1 subscriber moving to/from an area covered by another TBS but same DXT (handover) per hour
- A total of 104.46 milli-erlang of packet data transmitted in the TBS per hour
- 2 Automatic Vehicle Location messages sent per user every minute
- Additional to the previous traffic, a late entry signal is sent every second as a broadcast in the AI

Registrations

In the customer's network, a work day is divided in work shifts, every shift last 6 hours, for a total of 4 work shifts in one day, taking this into account, and by the fact that workers turn on their mobile radios at the beginning of their work shift, we can come to the conclusion that most registrations in the network happen in bursts of short time frames every 6 hours, this time windows last usually few minutes, depending on the people, and on the department those people will be working at.

In general, and by consulting with the system architect of the customer's network, we came to the conclusion that the majority of users turn on their radios and register within 5 to 15 minutes after their work shifts begin. Based in that information, we decided to have 3 basic time frames for our testing, one, being the most flexible, is when all subscribers register to the network in 15 minutes, another is when all subscribers register in 10 minutes, and the last one, which is the most dangerous, when all subscribers register within 5 minutes of the start of their work shift.

With the previous information, and considering the same criteria as explained in the background traffic section regarding the number of subscribers to be included in a TBS. I produced the following table:

<table>
<tr><td colspan="6">One Work Shift</td></tr>
<tr><td colspan="6">Signaling messages for 65 TR registrations</td></tr>
<tr><td>Case 1</td><td>Case 2</td><td>Case 3</td><td colspan="3">Traffic common for all cases</td></tr>
<tr><td colspan="3">Time</td><td>Protocol</td><td>Direction</td><td>#</td></tr>
<tr><td rowspan="5">First
3
Minutes</td><td rowspan="5">First
6
Minutes</td><td rowspan="5">First
9
Minutes</td><td>Location Update Demand</td><td>Up</td><td>55</td></tr>
<tr><td rowspan="2">Mutual Authentication</td><td>Up</td><td>110</td></tr>
<tr><td>Down</td><td>110</td></tr>
<tr><td>Accept</td><td>Down</td><td>55</td></tr>
<tr><td>Group Reporting</td><td>Down</td><td>NA</td></tr>
<tr><td rowspan="5">Next
2
Minutes</td><td rowspan="5">Next
4
Minutes</td><td rowspan="5">Next
6
Minutes</td><td>Location Update Demand</td><td>Up</td><td>10</td></tr>
<tr><td rowspan="2">Mutual Authentication</td><td>Up</td><td>20</td></tr>
<tr><td>Down</td><td>20</td></tr>
<tr><td>Accept</td><td>Down</td><td>10</td></tr>
<tr><td>Group Reporting</td><td>Down</td><td>NA</td></tr>
</table>

Table 2 - Registrations per work shift in a TBS for the testing

In table 2, we can see the 3 time frames in which we are going to base our testing; these time frames are marked as case1, case2 and case3. Each case is subdivided in 2 time slots, the first one is 3/5 of the total time, while the second one is 2/5 of the total time for all cases.
When talking to the system architect, we came to the conclusion that most of the subscribers (being 85%) register in during the very beginning of the times we had specified (our 5, 10 and 15 minute cases), thus, by his advice, I took the first 3/5 as the time when 85% of the users register to the network at the beginning of their work shifts, and the other 2/5 of the time, is used to register only the rest of the subscribers, being only 15%.
Because of this, we can see in the table that the total number of subscribers (65) is split in 2 amounts, 55 radios during the first 3/5 of the time, and 10 radios during the last 2/5 of the time.

In the right part of the table, I depicted the signaling messages used during the registration of the amount of mobile radios I explained in the previous paragraph. The regular registration protocol in the TETRA standard is, in a very simple way of explaining, as follows:

- A mobile subscriber powers up and sends a location update demand message to the base station in order to advertise that it is on.
- The base station then sends a challenge to the radio in order to authenticate it.
- The radio responds with the answer to the challenge.
- The base station sends an accept message and the communication can start.
- The base station then sends a report of the groups to be used in the radio

This registration's traffic explained above, together with the background traffic, is what we can expect at every beginning of a work shift in the customer's network, this is the basic load with similar conditions as in the customer network, and our purpose will be, to test on top of that, the OTAK traffic to be introduced when the E2EE solution is deployed. In order to do that, we need to define the OTAK traffic both during the implementation of the solution, which is when all equipment is migrated to use E2EE with over the air keying, and during the rest of the time, when regular registration and rekeying of radios will happen, but no initial configuration is needed.

OTAK

According to our collected customer's data corresponding to the network architecture and the expected delivery of keys for E2EE purposes defined by their crypto plan, I calculated the average OTAK messaging expected in the customer's network, in order to have a basis upon where our test plan and environment should be based.

There are 3 basic cases considered which define the main load of E2EE OTAK traffic, I explain them here below.

Initial Configuration or Deployment Phase

During the deployment of the solution in the customer's network, the TRs are going to be migrated to E2EE with OTAK capability little by little, with this concept, we can visualize that because few TRs are migrated per day, also the OTAK traffic will be increasing every day until the point that all Tetra Radios are using E2EE with OTAK capability. On top of that, we must consider that the OTAK traffic during the first registrations is much higher that during regular (already configured) TR registration, this is because during the first registration, the TR will receive all crypto configuration at once from KMF, hence creating a high load of traffic in the network.
Figure 20 shows the calculation of a 10 times load of expected OTAK traffic during the initial registration for 64 Smart Cards (the average amount of TRs expected per TBS in every work shift).

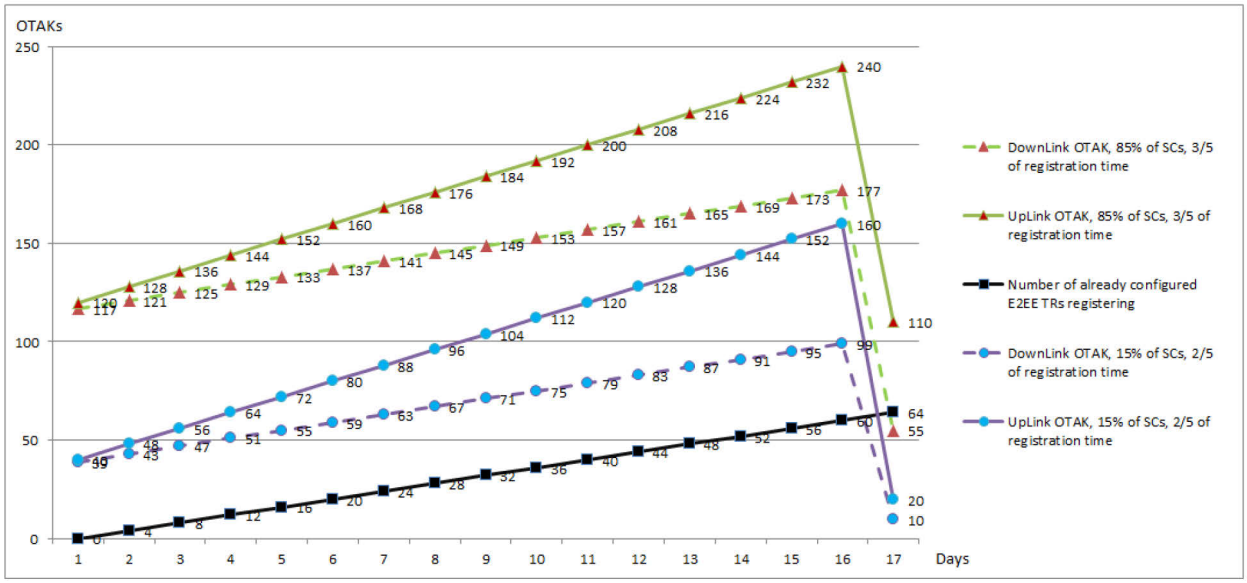

Figure 20 - A 10 times load of expected OTAK traffic per work shift in a TBS

Regular registration at every work shift

In the right part of the graphic in figure 20, we can see that the line drops significantly, that is the amount of OTAK traffic expected every time a work shift starts, when all TRs have already registered and there is no need to do an initial configuration, in this thesis, I call it “Regular Phase”, and it is explained in section 4.6.1.3.

Crypto period expiration

When a crypto period expiration happens, all Crypto Groups configured with that crypto period need to be renewed, to do this, the KMF will generate new random encryption keys prior to deliver them to the TRs containing the mentioned CGs.
I assumed that for better resilience, all CGs within a given KMF will be expired at the same time, which will cause the generation and delivery of keys to all Tetra Radios served by the mentioned KMF.

The crypto period expiration OTAK calculations can be divided into 2 main cases, one being the delivery of OTAK messages to current online TETRA Radios (the ones currently in service during the work shift), and the delivery of OTAK messages to offline TETRA Radios (not actively online during the moment of key generation, but which will be updated at the time they come back online).

Table 3 shows the amount of OTAK messages expected to be sent to the average amount of online TETRA Radios present in a TETRA Base Station during a given work shift if the total amount of CGs is renewed at the same time (the number of CGs is defined by the customer’s crypto plan):

Crypto Period Expiration OTAKs sent to the currently online TRs		
Smart Cards		
64,28	~ 65	
	DownLink OTAK	UpLink (Acks)
	520	520

Table 3 - OTAK messaging sent to online radios after a crypto period expiration

For the case where the OTAK messages are sent to the TETRA Radios which are offline at the moment of key renewal, the calculations need to be specified in a slightly different way, because as explained in section 4.6.1.3, the TRs are not expected to come at once online, and for that reason, I specified a 85% and 15% of registration traffic to happen in 3/5 and 2/5 of the time respectively (for more details see the corresponding test case scenario in section 4.6.1.3).
Table 4 shows the calculated amount of OTAK traffic for the specified percentages of TRs and duration of the registration.

Crypto Period Expiration OTAKs sent to TRs next time they go online (includes registration)				
Smart Cards	85 % out of total SCs		15% out of total SCs	
64	54,4 ~ 55		9,6 ~ 10	
	DownLink OTAK	UpLink (Acks)	DownLink OTAK	UpLink (Acks)
	495	550	90	100

Table 4 - OTAK message traffic for key renewal when TRs become available

The following section explains how the given amount of traffic described in this load profile is used for the planned test cases during the verification phase in Helsinki's laboratory.

4.7.1.2. Lab network for AI load testing

In figure 21 is shown the laboratory network we built to support the required amount of traffic specified in the customer's network.
The figure does not show all the elements in the network, non-relevant elements (such as monitoring computers or CDD server) have been left out for the sake of simplicity and because they do not affect directly the flow of traffic concerning the E2EE solution; only the network elements directly affecting E2EE are shown in the figure.

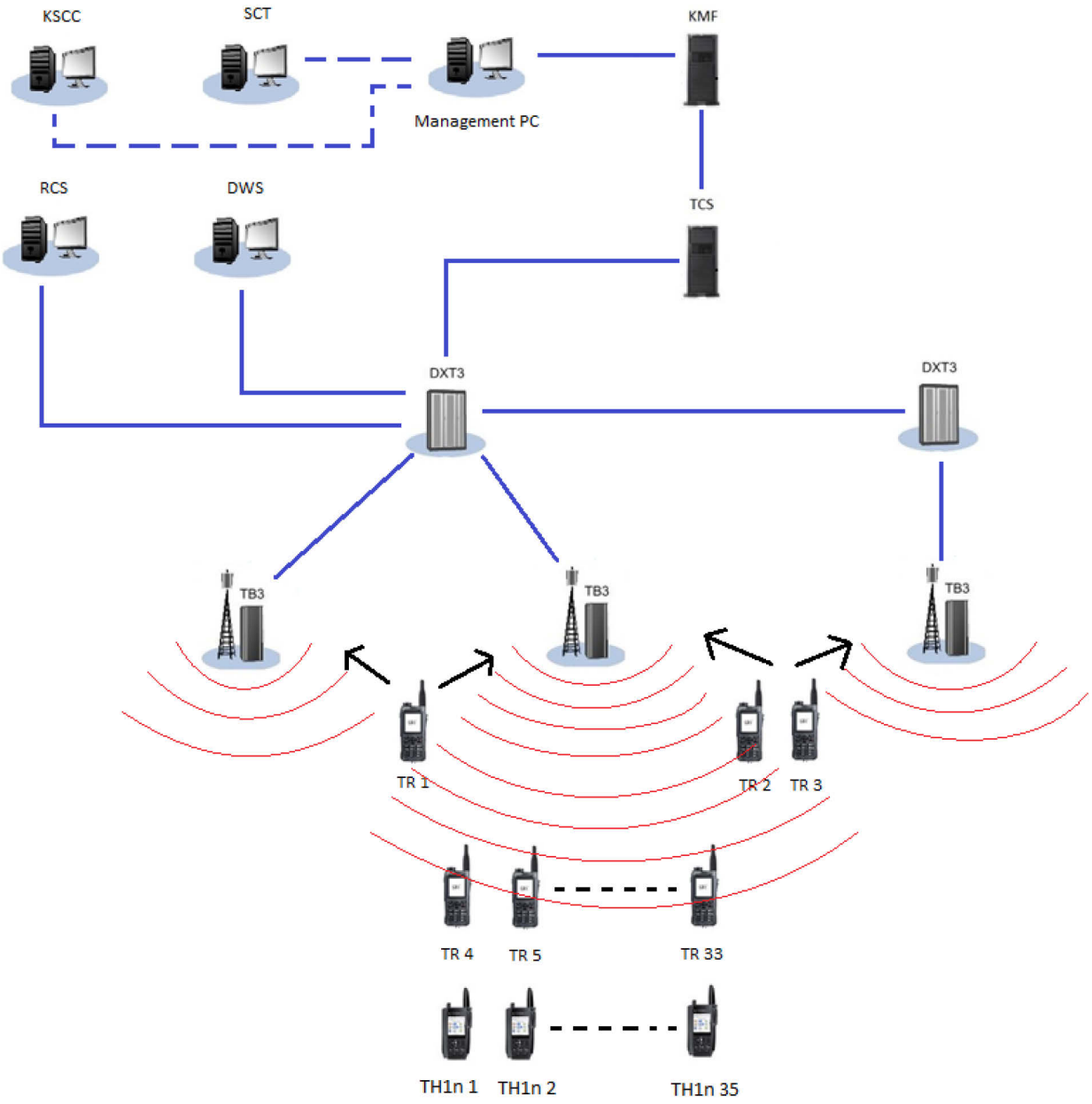

Figure 21 - Laboratory network for Air Interface testing

I am going to explain shortly the purpose of each element in the laboratory's test network:

- Management PC: At the top of the figure we can see this device which is a Windows 7 computer used to do general management of the E2EE elements in the network, inside of it is installed the KSCC and SCT (explained in sections 4.4.1.2 and 4.4.4.2) in order to manage Smart Card's configurations, it also has a remote connection to the KMF in order to do operations in KMF's GUI or to debug found defects from the log files.
- KMF: It is remotely IP connected to the management PC and connected to the TETRA network (DXT) through the TCS API (explained in section 4.4.2) in order to deliver E2EE key material to the intended recipients.

- DXT3s: their purpose is to switch the voice and data packets in the TETRA network, it is connected directly to the TETRA Base Stations which in turn transmit the mentioned packets to the end users (Radio Terminals), it is also connected directly to the RCS and DWS (dispatcher workstations) as well as to the TCS server (which acts as a bridge between the TETRA network and the outside network/KMF)
- RCS and DWS: In this specific test network, these dispatchers have the following purpose. RCS is used to receive group voice calls; normally, we would need at least 2 TRs to create a group call, however, because RCS can have several group calls open at the same time, we only need TRs as senders and doing so reduces the amount of required TRs for background traffic. DWS is used to do management operations inside of the TETRA network (such as adding subscribers, modifying organizations, etc.)
- TBSs: there are 3 TBSs in the network but only one is monitored for our tests, the other 2 TBSs are used for handover and roaming purposes (for background traffic)
- TETRA Radios: We can see that there are 2 different types of TRs, one of the groups has label “TH1n” and the other one has label “TR”. In SLC’s systems, only TH1n radios support OTAK E2EE, and due to the fact that they would be switched on/off frequently they are used with the sole purpose of exchanging OTAK signaling with the KMF. The other type labeled “TR” is an old model which we used only to create background traffic (calls, messages, handovers, etc.)

4.7.1.3. Test plan and test coverage definition

The Air Interface testing covers the analysis of the signaling generated by the E2EE OTAK messaging traffic over the Air Interface to find any bottlenecks and other possible defects generated by the newly introduced solution. This is accomplished by saturating the AI capacity with peak loads of traffic and analyzing its consequences in the behavior of the devices involved in the OTAK traffic flow.

In order to perform this task, I designed a test plan for the different uses of E2EE OTAK traffic expected in the customer’s network, and focused especially in the cases where high loads of OTAK traffic is expected at one given time.

Figure 22 shows my concept of E2EE OTAK testing over the Air Interface. The figure depicts 28 test cases, which form the basis of the E2EE OTAK testing over the AI. These 28 cases are divided in 4 testing sub-divisions, which correspond to the most critical phases of expected OTAK traffic in the customer’s network.

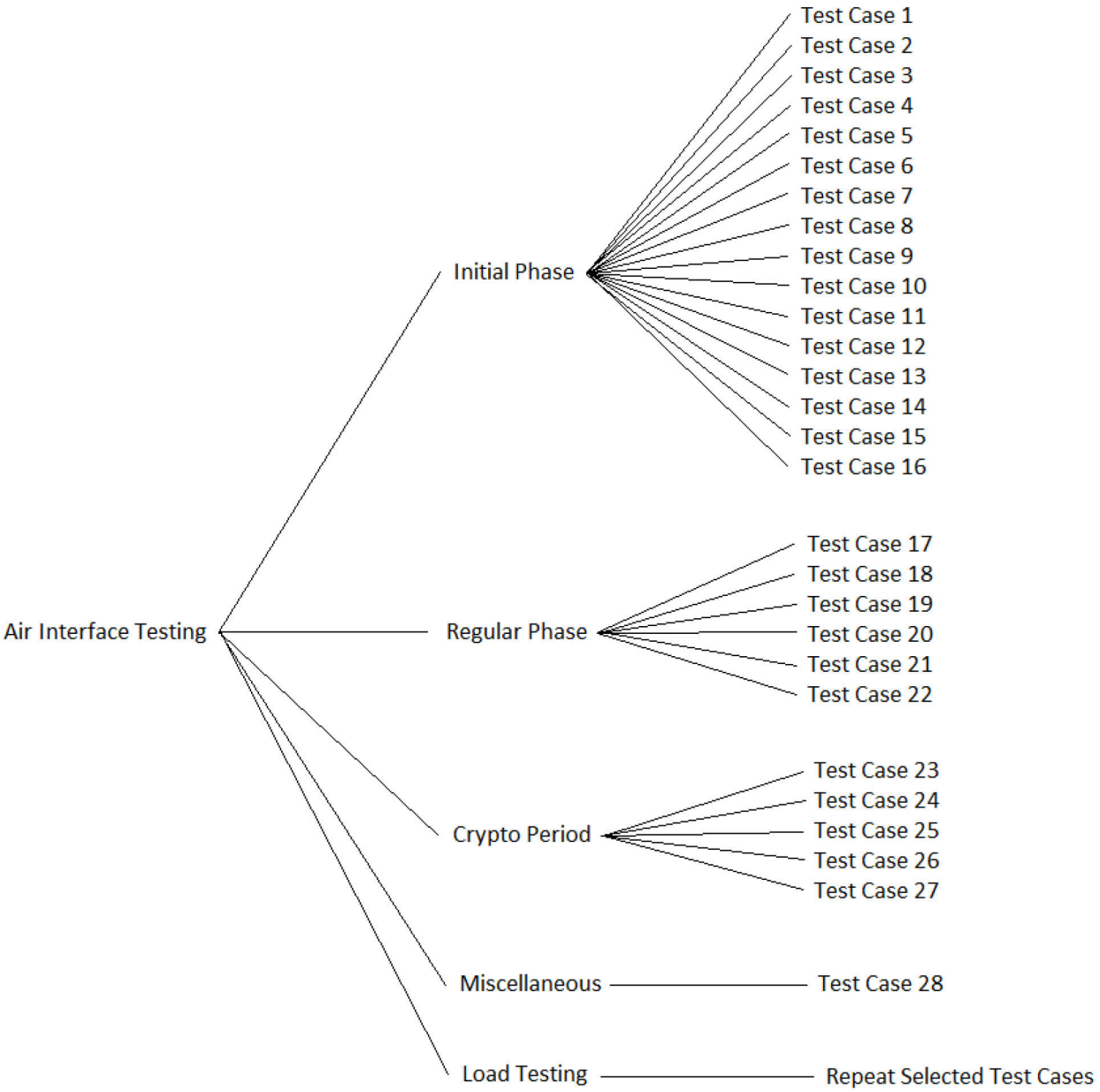

Figure 22 - Air Interface testing distribution

We can see in figure 22 that the sub-divisions of the AI testing correspond to the different calculations of OTAK traffic presented in section 4.6.1.1.

In the first sub-division of the air interface testing (Initial Phase) there are 16 test cases, and from the fact that in the customer's network, the average number of SCs present in a TBS at a given one time is 3.5, we can conclude the results in table 5:

Initial configurations per work shift per TBS				
Smart Cards	85% out of total SCs		15% out of total SCs	
3,571428571	3,04 ~ 3		0,535714286 ~ 1	
	DownLink OTAK	UpLink OTAK	DownLink OTAK	UpLink OTAK
	117	120	39	40

Table 5 - Number of UL and DL OTAK messages taken into account for the Initial Phase test cases

In order to explain table 5, it is necessary to state the following points.
According to the customer's network traffic information collected from the system architects:

- Every DXT serves an average of 58.3 TBS
- Every DXT serves an average of 15 000 SCs from TRs, which divided by the 58.3 TBS gives an average of 257.28 served by one TBS in one DXT.
- There will be 1000 Smart Cards migrated in the whole network to E2EE with OTAK capability per day
- There are 12 DXTs in the customer's network
- 1000 SCs migrated to OTAK per day in the whole network means that 83.33 SCs will be migrated in average per DXT
- 83.3 SCs migrated per DXT divided by 58.3 TBS operating under a DXT tells us that 1.42 SCs will be migrated per under a TBS per day
- There are 4 work shifts of 6 hours in the customer's network
- We have an average migration of 1.42 SCs under every TBS served by a DXT per day. That number of SCs is divided by 4 work shifts, which gives us 0.35 SCs in average.

➔ 0.35 SCs migrated under a given TBS per day is not enough to test the AI capacity for its breaking point, because of this; I decided to augment that number 10 times, in order to have a more representative amount of traffic. This gives us the number seen in table 5 under the column "Smart Cards". Although calculations based on average amount of traffic are not the best to dimension the traffic load, it was the only traffic information available in order to produce the given calculations.

According to the network architects, it has been seen in the customer network that at the beginning of a work shift, most users (roughly 85%) register to the network, while the rest (roughly 15%) register a short time after all the rest. This forms the basis of the numbers shown in table 5, where 3 SCs' (a rounded number corresponding to 85% of the TRs) OTAK messages are separated from the other 1 SC's (rounded from the remaining 15% of TRs) OTAK messages.

Because we are testing the behavior of the solution over the AI, there is no need to analyze the traffic from more than 1 TBS, which means that from figure 21 (network test environment) we only need 4 TH1n radios at any one time, however, for resiliency purposes, and because we had 35 TH1n radios available when testing the TETRA Radios' side of the solution (testing performed by Jyväskylä's

engineers, hence, not included in this thesis), then we decided to use all 35 TH1n radios to increase the load of the network and aim to reach peak levels.

Initial phase testing

This test phase is divided into 16 test cases because considering that 4 SCs are migrated to OTAK during every work shift, and that it happens 4 times in 1 day, then it would take us 16 days to migrate all the 256 SCs served in average per TBS in the customer's network. In reality, the migration phase will take about half a year because the number of Smart Cards is not 4 per TBS, but 3.57 divided by 10 (as specified in the points above), however, for testing purposes, and to test the system more comprehensively, this supposition is useful in our case.

The test plan document produced by myself explains in detail the steps to do in every single test case, however, for simplicity and convenience, I will not write in this thesis all the procedures, just the main ideas of them.

Figure 23 below is an edited version of figure 20 from section 4.6.1.1. Here each "Day" corresponds to a test case, Day 1 starts with 0 configured OTAK E2EE TRs, so I assumed that every day, during every work shift, 4 TRs would be migrated to OTAK E2EE, that is why the line "Number of already configured E2EE TRs registering" shows numbers increasing in intervals of 4, I also assumed, that the users of the TRs will remain in their own work shift, hence, there will be no user moving from work shift 1 to work shift 2, that is to say, by day 2, work shift 1, the only radios already configured for OTAK E2EE will be the ones migrated in day 1, work shift 1, and not anyone from any work shift from 2 to 4.

In the figure we can also see that besides of the already explained division of users in 85% and 15%, there are 2 lines marking a division in time as 3/5 and 2/5 (which means first 3, 6, 9 and next 2, 4, 6 minutes, equaling 3 sets of times), this time frames were decided together with the system architect of the OTAK E2EE implementation, who gathered statistical information from the customer's network and came up with these times for user registrations to the network.

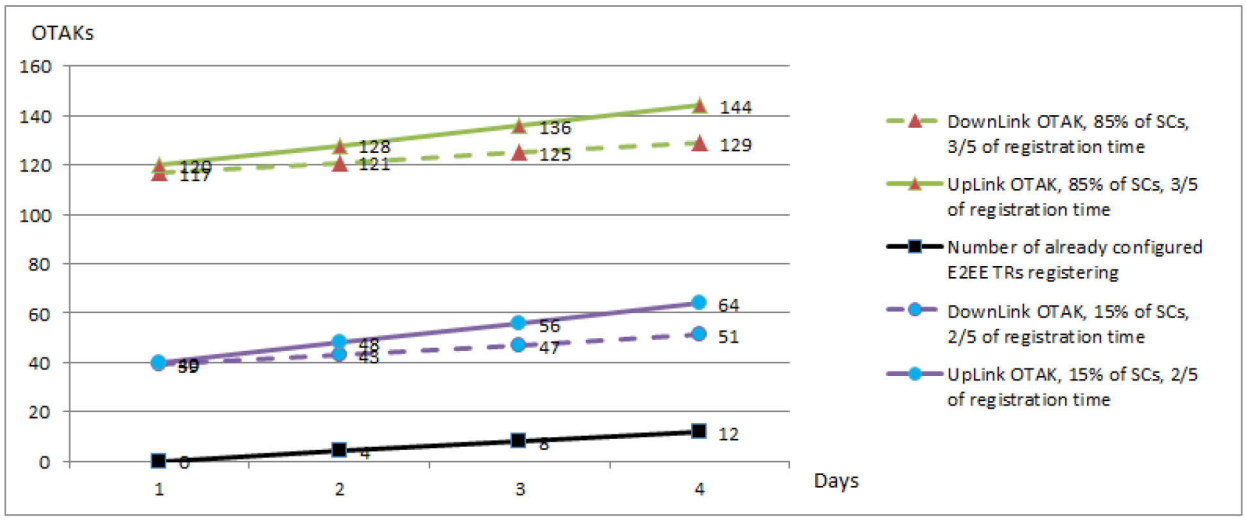

Figure 23 - Number of OTAK messages corresponding to the 4 first test cases of the Initial Phase

We have 16 test cases in figure 22, however, the times corresponding to figure 23 should be read as 3 sets of time, one set of time such as 'First 3 and Next 2 minutes', another set of time as 'First 3 and Next 4 minutes' and the last set of time as 'First 9 and Next 6 minutes', to come to the conclusion that we need to run this test 3 times, with the different sets of time frames mentioned here, for a total of 3 scenarios per test case (totaling 5, 10 and 15 minutes respectively). The reason for this is to see the behavior on the network when the traffic is applied in different time frames, and because in real life, it is impossible to have all the users turning on their radios every day at the same time, which would make little sense to only test one time scenario per test case.

The "OTAK" numbers of figure 23 show the number of expected messages (UL and DL) exchanged between all the TRs in one TBS and the KMF. As seen, the quantity of UL OTAK messages is higher than its DL counterpart, this is because in the OTAK signaling protocol there is one extra message going in the UL direction. For example, in "Day 1", for the line corresponding to 3/5 of time (3, 6 or 9 min) there are 117 DL and 120 UL OTAK messages, which correspond to the initial registrations of 3 TRs to the KMF (the extra UL message makes the UL quantity 3 numbers higher than the DL) with the applied crypto plan.
Due to confidentiality reasons, I am not allowed to explain any further the basis on which I calculated the OTAK messages exchanged by every TR during its registration. However, I can say that the amount of OTAK messages exchanged is the result of the crypto groups and associations defined in the crypto plan upon which the KMF is configured, in our laboratory case, the amount is 39 UL and 40 DL OTAK messages per TETRA Radio.

Regular phase testing

This section covers the tests performed taking into account the regular conditions of operation of the E2EE solution once the migration (or initial) phase has been accomplished.

For this test phase, we designed 6 test cases, I mention “we” because for this phase, due to his expertise in the subject, the test engineer in charge of the TETRA infrastructure part (DXT and TBS) designed test cases 19 to 22, while I designed only 1 of them, and we both designed the remaining test case together.

For simplicity and convenience, I will not explain in full detail the test cases, however, I will mention their purpose and their most important points in the following list:

- Test Case 17: The purpose of this test case was to test the behavior of the network when users register at the beginning of a work shift in every day operations, due to the fact that when users turn on their TRs, the TR performs the registration protocol to the KMF and OTAK messages are exchanged. This was my own contribution to this phase (Regular Phase) of the test plan.

- Test Case 18: The purpose of this test case was to test the result of an end-user action on the KMF’s GUI during regular operations. In our E2EE solution, a network administrator can order a renewal of encryption keys through the KMF’s GUI, the test consists of renewing the keys of a Crypto Group, while users are using the TRs in a regular way during a work shift. This test case was a joint contribution by me and the experienced test engineer managing the TETRA Air Interface.

- Test Case 19: The target of this test case was to prove that key loading using OTAK works properly when users are in an individual call. In essence, the procedure was to create individual calls before a crypto period expired, and end those calls once the key load to the TRs had finished.

- Test Case 20: In a similar case to test case 19, this test case ensures that key load from KMF to TRs works fine in case of a call, however, this time the call is a group call. The procedure in this case was to divide the amount of radios in groups and then make a group call in one them before the crypto period expiration, then wait until the keys have been loaded in the TRs and check for errors or any other inconsistence.

- Test Case 21: This case is similar to test case 20, the difference is that 2 TBS stations are in use, hence, less traffic is going in the downlink direction of the TETRA Air Interface while the group calls are happening. The main purpose of this test case was to check the correct operation of the key loads on a different traffic setting involving one way direction of the voice speech.

- Test Case 22: The last one of the test cases for this Regular Phase, was to check that the key loads from KMF to TRs work properly when the radios go on and off a call at the same time that keys are being loaded. For this purpose, group calls were used.

Crypto period testing

This section mentions the expected situations in the customer's network when the encryption keys expire and there is a need to deliver new keys to the TETRA Radios.
The situations in the 5 test cases comprising this section are closely related to the *Regular Phase,* however, because the crypto period expires only at determined times, the traffic presented during that expiration is not precisely affecting the day to day operations most of the time, for which I decided to isolate those test cases within their own section.

I will give an overview of the 5 test cases included in this section of the test plan, as well as mention their main purpose. However, for simplicity and clarity, I will not explain in detail their steps.

- Test Case 23: The purpose of this test case is to get a baseline of the behavior of the distribution of keys for a crypto period change with ideal network conditions (empty network and no fading), so then we would compare this behavior with other test cases' behaviors.
  We had all our TRs up to date and KMF configured correctly, then we waited until the crypto period expired and analyzed the traffic in the network.

- Test Case 24: The target of this test case is to verify that the traffic introduced by this solution when the crypto period expires would not affect the regular operations in the network.
  This case has the same procedure as test case 23 with the difference than instead of having an empty network, we had regular traffic (voice and data) happening in the network, as well as fading conditions in order to make it more challenging. This test case was one of my contributions.

- Test Case 25: This is a variation of test case 24, in this case, we slightly modified the scenario, so when a crypto period expires, instead of sending the new keys to the currently online TRs, we decided to have them offline, so they would receive the updates on the keys when they register again, the main purpose was to see the behavior of the network with the traffic load produced by the crypto period expiration buffered in the server and released at the registration of the TRs. In order to challenge the capacity, we switched on all radios at once. This test case was one of my contributions.

- Test Case 26: The target of this test was to verify how a user-initiated action affected the crypto period change and delivery of keys.

For this test, we re-keyed few TRs in KMF's GUI just before the crypto period expired, in order to see if those actions triggered any errors while the crypto period was generating new keys.

- Test Case 27: This test case focuses in 3 main aspects of a radio network performance. The first target was to verify that KMF can adapt the OTAK traffic to a variable amount of background traffic. The second purpose was to ensure that OTAK downlink messages work even if there is fragmentation of packages. The last purpose focuses in verifying that uplink messages are correctly handled even if there are repetitions of the same ACKs sent due to collisions when competing for access to the radio network by the TRs.
  For this test we had variable background load and fading conditions running, we waited that the crypto period expired and analyzed the traffic.

Miscellaneous test cases

In this section we had only 1 test case defined, this test case verifies that the KMF is capable to manage the delivery of keys also when there is a failure on the wired network connecting it with the DXT (through TCS).
The procedure is to make the TCS unavailable to the KMF during a period of delivery of keys, and verify that it resumes the procedure without errors when the TCS was reconnected.

KMF has the capability to connect to many TCSs through different NICs to use as alternate routes when one network link fails, however, that feature was tested during product verification in France and it was not part of our scope in Helsinki.

Besides of our test case (joint contribution by me and my colleague test engineer), we left this section opened for probable future test cases to be added.

Load testing

The sole purpose of this testing section was to repeat the most challenging test cases with higher amounts of background traffic, or by increasing the amount of CGs and associations in the KMF's configuration.

Not all tests are to be repeated, only the ones that gave more interesting results.

4.7.2. AI testing phase

In this section I will explain how and when the tests over the Air Interface were performed, as well as the defects found and the procedure to correct them.

4.7.2.1. Test plan execution

We starting to execute the test plan in the beginning of October 2015, after I got the knowledge corresponding to KMF's installation, configuration and maintenance, and we configured the planned test environment.

The first testing round lasted until beginning of December. In this testing round we found the most important errors and submitted them for correction to the corresponding teams.
After the defects were submitted for corrections, there was a second round of testing, which yielded more defects, as well as verification of some corrections.
At the moment of writing this thesis, the cycle of defect finding and corrections applied is still going on.

There were 2 experienced test engineers and me participating in the project.
One test engineer (from jyväskylä's team) was in charge to find defects for the TETRA Radios model TH1n, which are in use for OTAK key management. He reported the TR's defects directly to Jyväskylä using their own defect management tool system.
The second test engineer was in charge of analyzing the traffic over the Air Interface, he is an experienced colleague with many years of experience in its field and he was the one leading the hands-on testing as well as in charge of raising the defects concerning the TETRA network (DXT, TBS, etc.).

My role in the project was to manage everything concerning the E2EE on terms of key management in different parts of the system (KMF and TRs' SCs). My main responsibility was to install, configure and maintain the KMF, find defects concerning its operations, submit the defects to the corresponding team in France, and follow the corrections of those defects in order to solve the issues and deliver a mature product to the deployment team.

During most part of the testing, we all had to debug the information together in order to come up with a conclusion. For example, after performing a test case, I had to check the logs in the KMF in order to find out what was the exact set of events, and depending on the case, what and where the errors had occurred inside of the server. My colleagues, complementarily, had to analyze logs in the DXT-TBS, and TRs correspondingly.

4.7.2.2. Defects

There were 24 defects found by our team in Helsinki during the first round of testing, which lasted until December 2015, some of those defects are very visible to customers, while some others are not so visible and require specific conditions to happen, but when they happen, are disastrous to the network, yet some other defects, are small and less important than the rest, however, they would need to be corrected in the future.

I will enumerate in the tables below the defects in order of importance, starting from critical or major defects in table 6, passing through the medium-importance defects in table 7, and finishing with the less important defects in table 8.
For convenience, I do not list all of the 24 defects found, since some of them have not enough relevance to be mentioned and also, to add simplicity to this thesis, I decided to leave those out.

Major or critical defects

A more comprehensive explanation of a given defect is provided after the table if required.

	Severity	Major	
	Problem	**Description**	**How it was found**
1	Backup operation interferes with delivery of OTAKs	Every day there is an automatic backup at 1 am. If the crypto period expires at the same time or during the time that the backup is in progress, KMF stops operating and doesn't deliver OTAK messages anymore	We need a short crypto period for testing, so I configured it to happen every hour sharp, because the instance of 01:00 collided with the backup, this failure happened
2	Crypto module failures	KMF's crypto module (MGEM + SC4KMF in conjunction with some tables in the server's DB) raises an alarm indicating that a SC has failed with a fatal state and turns the crypto module indicator light in GUI red. We identified 2 scenarios when this happens: a) When the keys for a given crypto period have not been delivered yet (because of congestion in the network) at the time a new crypto period begins. b) When a SC's registration is unsuccessful	a) This point was discovered when besides of the background traffic, we introduced fading in the radio spectrum of the AI, that originated that the keys were not always delivered in time before the next crypto period started. b) This point was discovered when we introduced RCS into the network.
3	Wrong sequence number sent at every OTAK delivery retry	When KMF does not receive a positive acknowledgement to an OTAK message, it retries to deliver it after a timeout expires. In this case, the sequence number was not increased at the retry attempts. This causes problems because sometimes the TR had already acknowledge the message but it had not arrived to the KMF (because of congestion in the network) before it retried, however, the TR was already expected an increment in the sequence number (no matter if was a retry of a past OTAK or a new OTAK message)	This error came into light when we analyzed the signaling traffic with high load over the AI, by using background traffic and fading conditions, which caused a lot of random access collisions.
4	Link Layer Ack arriving after Application Layer Ack	In perfect conditions, KMF expects to receive first a Link Layer acknowledgement before receiving an Application Layer acknowledgement (either positive or negative), however, when congestion in the network is present, the Application Layer acknowledgement might arrive first. In this case, KMF sent a OTAK #2 (App Layer message) upon receiving a positive OTAK Ack #1, however, it then received a negative Link Layer #1 Ack, and it resent OTAK #2 because it thought that it corresponded to the Link Layer # 2 Ack. This point is explained more deeply below.	This was in fact one very difficult problem to debug and understand, it required high load over the AI, and deep analysis in all KMF, DXT, TBS and TR's logs in order to figure out the exact flow of traffic going on when the problem ocurred.

Table 6 - Major problems found during E2EE testing over AI

Major problem #4

I will explain here the point number 4 mentioned in table 6.

The flow of expected OTAK traffic in perfect conditions from KMF's point of view is illustrated in the following image:

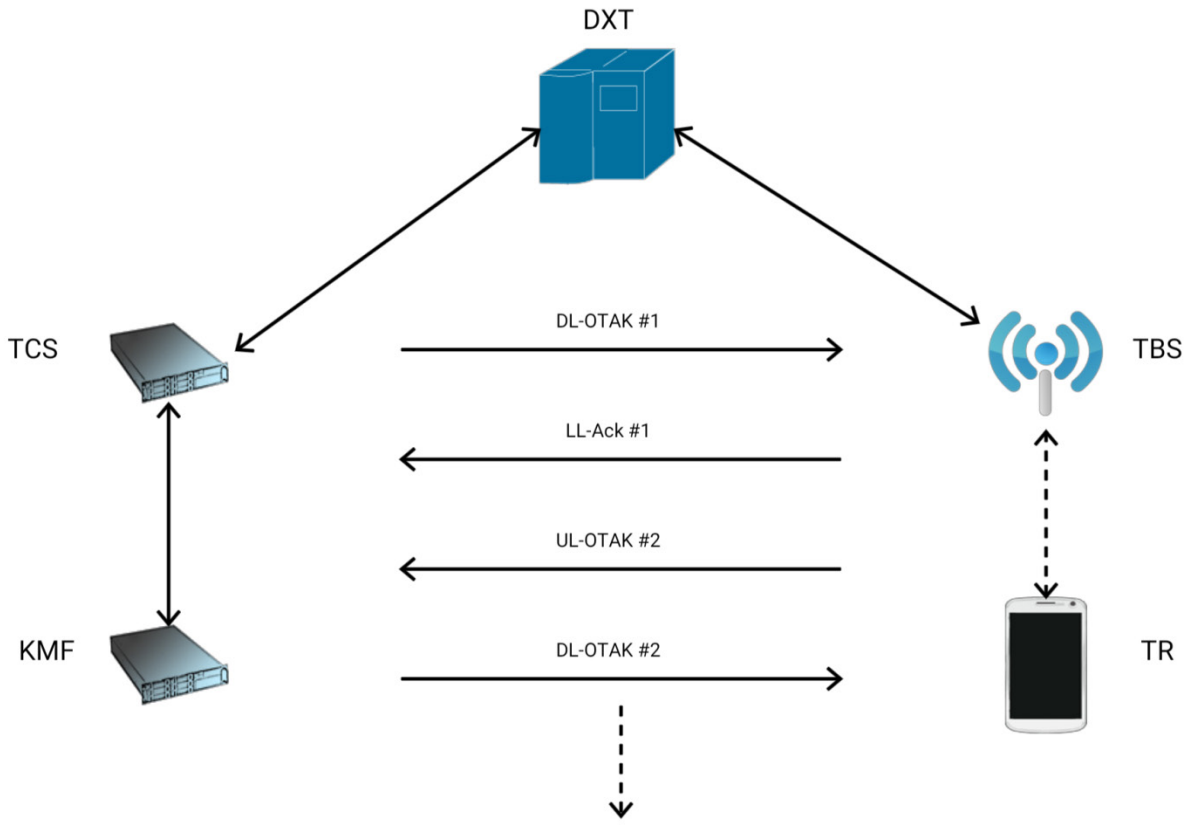

Figure 24 - OTAK traffic flow in perfect conditions

In figure 24, I illustrated the flow of OTAK traffic in a very simplified manner, there are 2 layers involved in the exchange of messages, an underlying layer called Link Layer (LL in the figure), and the application layer (OTAK).
When KMF sends a DownLink OTAK message, it expects first a LL Acknowledgement and then an UpLink OTAK acknowledging the DL-OTAK received by the end user.
Figure 24 only shows the expected traffic flow in perfect conditions, however, the traffic flow does not always happen in that order.

During our tests, we found out that due to network congestion (mainly in the AI), sometimes the LL-Ack arrived later than the UL-OTAK to the KMF, and because KMF was not prepared to receive such a flow of traffic, an error occurred.

In the example written in table 6, I give numbers to the OTAKs in order to make the idea clearer.
KMF first sent a DL-OTAK, I gave to this message the number 1, then it received the corresponding UL-OTAK #1 (LL-Ack is missing due to congestion), after that, it continued immediately to deliver the next DL-OTAK (#2) for the same subscriber, however, the missing LL-Ack was not mapped to DL-OTAK #1, so when that LL-Ack was received late (and it was negative), KMF thought it was meant for OTAK #2 and

because it was negative, it concluded that OTAK #2 had failed and retransmitted DL-OTAK #2 erroneously.

Medium importance defects

Table 7 shows the medium importance defects we found during our testing in Helsinki, these errors would not cause that the network or the E2EE solution stops working in most of the cases, however, some errors would cause it if the conditions in which they are produced happen.
After the table, I explain in more detail the most important elements from the list.

	Severity	Medium	
	Problem	**Description**	**How it was found**
1	KMF does not execute commands when set to a wrong time zones	In the algorithm of operation of KMF, there is a step in which it compares its computer time to the times inserted into the commands queued in its DB, in order to send them at the right time. Because the commands in queue are generated with GMT time, when the server was configured with a negative time (east of GMT), KMF failed to send the OTAKs due to an exception.	During product testing, the French team always installed KMF with the French time zone. In Helsinki, I installed it with american time zone
2	When the amount of logs is huge, it takes too long to display in GUI	There is no way for the operator to specify what portion of the logs saved in DB wants to see displayed in GUI. So when opening the logs tab in GUI the full log table from DB is fetched, causing the GUI to be unoperable for a determined amount of time	After some weeks of testing, the log table in DB became too big, and eventually the time to fetch them was too long
3	Useless activation of key OTAKs sent over the AI	Every time a crypto period expires, KMF generates a key for the CG which expired, a creates a job in DB for every SC to which that key must be sent. If a given SC (ie. a TR put in storage for few months) is away from the network for several crypto periods, KMF keeps the jobs to activate the keys in its DB, and sends all those activation OTAKs to the SC when it comes back online, this causes unecessary load in the AI taking the bandwith that could be used for other traffic	Because the crypto period is configured to 1 hour in our lab, and some times we left few TRs offline overnight or on weekends, we noticed the huge amount of traffic when we wanted to use them again
4	Manual key requst from TRs take too long	In our TETRA system, if for some reason a user is not able to communicate in a given talk group due to encyption problems, he/she can request to get his/her TR's encryption keys renewed using his/her TR's menu. During that procedure, there is one step in which the KMF takes too long to process an OTAK message and because of that, the user holding the TR has to wait to long and may even cancel the operation.	Debugging other problems related to TETRA Radios, Jyväskylä's team test engineer discovered this problem which then I reported as a KMF problem
5	All OTAK messages are sent with priority "Important"	The recommendation for TETRA standards especify that there may be different levels of priority assigned to SDS messaging, this priority may be assigned in the TETRA network (DXT) or in a system funtioning as a third party application behind a TCS (such as KMF). In our case, KMF was marking all OTAK messages as "Important", hence, taking more precedence than other more important signaling traffic on the AI.	My collegue, experienced TETRA test engineer, discovered this when he was trying to tune the OTAK traffic over the AI through the DXT, noticing that all OTAKs coming from KMF were already set to the "Important" status.
6	MGEM authentication failure	After installing all necessary software in the windows server, and when trying to either start TKM service (deamon for KMF), the start up fails indicating an authentication failure between the SC4KMF and MGEM.	This was discovered when trying to run KMF for the first time.
7	Waste of bandwitdth by useless OTAK messages	When KMF sends an OTAK message to a TR, and it does not receive an acknowlwdgement confirming the reception of the message, it uses a timer to wait for an especified amount of timer before trying again. In this case, KMF was resending the OTAK message no matter what answer it received, even if the DXT was answering that the TR had been switched off manually and it's not reachable.	We noticed the attempts to deliver the OTAK messages to TRs we left switched on during the night or weekends.
8	Operator commands in the GUI take to long to initiate	An KMF's operator can issue commands to delete all crypto configuration from the TR's Smart Card, as well as reload all the current crypto configuration. This commands were not set with the right priority in the queue of jobs performed by the KMF, so instead of starting right away, some other less important jobs were executed first (ie. the load of a future key)	We noticed this when we did manual operations on the GUI

Table 7 - Medium importance defects found in Helsinki

Point #1
This error is not very probable to happen in the customer network, first, because the computer must be in a negative time zone (compared to GMT), which usually means a server in America, and the second reason, because the one who installs the DB is usually an engineer from SLC in the process of deploying the solution. However, if the DB would have been set in a server with a negative time zone, the KMF would simply not work, and no encryption keys would be delivered to TRs.

Point #3
The time of validity of an encryption key for communication (TEK) depends on the customer's crypto plan, it could vary widely from few days to few months. It is not so probable that a couple of TRs would be away for many crypto periods and they would be switched on in the same cell during the same time, however, not being probable doesn't mean that it would not happen, when it happens, the load on the AI would be huge, especially if the number of expired crypto periods is high, this could cause that important group or individual calls do not get the resources they must have.

The rest of the points in table 7 are self-explanatory.

Minor defects

Table 8 shows some of the minor defects we found during our testing. They are not so important and are not to be included in the first version of KMF, however, they should be taken into account for version 2.

	Severity	Minor	
	Problem	**Description**	**How it was found**
1	No mean to delete logs from the User Interface	In the KMF's GUI, there is no button or any other mean to delete the displayed logs, this makes hard to debug some problems because there is need to filter among more data	It is evident in the GUI, after many logs have been recorded, I wanted to get rid of the old ones
2	No way to delete crypto configuration once it has been imported	Because this is the first version of KMF, and because a network is intended to work with a pre-defined crypto plan, KMF was designed tailored to the current customer, and no option is available to the administrator to delete crypto configuration (ie. Crypto Groups) once they have been imported	For testing purposes, I wanted to modify the crypto configuration, then I found out KMF is not flexible enough to do this
3	No option to shorten a User Group Range	As point 2 describes, once data is imported, there is no option to delete it. In this case, a User Group Range of ISSIs could not be shortened, which causes problems if we want to use some ISSIs from that User Group into another one with different CGs	I wanted to separate the TRs into 2 groups by including them in different User Groups, so they would receive the encryption keys belonging to different CGs

Table 8 - Minor defects found during the OTAK E2EE testing over the AI in Helsinki

Table 8 concludes this section regarding the description of defects, in section 4.6.2.3 I will discussed the cause and solution of the defects (from a tester's point of view), as well the round 2 of our testing performed in Helsinki.

4.7.2.3. Results, benefits and solution of found defects

The project resulted in a quite extensive work which took more time than anticipated. Many defects were found, some of them very critical, which would have degraded or completely crashed parts of the customer network in case they would have happened in the field.

The testing over the Air Interface was, in my opinion, a critical part of the End-to-End Encryption solution's validation, due to the fact that the Air Interface is the bottleneck of traffic of any wireless/cellular network because of the physical characteristics and utilization of the radio spectrum.

The results of this verification work yielded a better working solution with a more mature product (KMF) in its core, which greatly benefits Secure Land Communication's portfolio overall and validates the correct operation of new features (Over The Air Keying) to an existing End-to-End Encryption solution, upgrading in this way the capacity and performance of encryption key delivery to end users compared to previous methods of delivering the keys in question. This benefits not only the company and our systems portfolio, but also our customers and their personnel, which now will have a more automated and easy way of distributing the encryption keys needed for their daily communications.

At the time of finishing this thesis, most of the defects mentioned above have been solved, special attention was taken towards the critical ones, these defect corrections have been implemented in several instances of software. We, our testing team in Helsinki, worked closely with the TETRA Radio terminal team from Jyväskylä and the KMF's development team from France, having continuous communication over email, phone and workshops were all teams were working physically together. This cooperation allowed us to develop corrections and test them in several iterations of software for both KMF and TR sides (the rest of the network didn't suffer considerable changes)

Starting from January, we have been testing subsequent improvements in KMF's software, improvements that besides fixing some of the defects found before December, have also introduced some new errors.
From January 2016 until the time of writing this thesis, 12 new defects have been found, from those 12 defects, 3 have already been fixed, and from the remaining 9, the major ones are expected to be fixed by the next iteration of the KMF software.

KMF, at this point in time, still have issues, but it also has improved significantly, I discuss this in sections 6 and 7 later in this thesis.

5. My contributions in a nutshell

My role during this project was mainly to be the person in charge of the Key Management Facility server in Helsinki, taking care of installing it, configuring it, keep it working, finding its defects, analyze those defects, report them to the development team, follow the defects' solution process, and test the corrections for those defects in order to ensure that there are no critical errors that could affect the traffic on the customer's network in terms of the Air Interface's signaling capacity.

The testing was divided in different geographical areas depending on what parts of the system were to be tested. In Helsinki, we tested 3 parts of the system, one was the TETRA Radio terminals, the second was the DXT-TBS connection, these parts were tested by 2 test engineers, and the KMF was my responsibility.

In order to perform my job, I first learned the basic theory of how End-to-End Encryption in SLC's TETRA networks work, for that I read different company papers and part of the SFPG recommendations whitepapers. The second step was to be trained in the technical details of the KMF itself, for that I spent some time in France with the team in charge of testing the KMF on a product level, then I continued experimenting myself in Helsinki by installing and configuring our own KMF servers.
Lastly, I participated in the system level testing together with the test engineers assigned to this project.

Another of my important contributions to this project was to write the test plan and test strategy which depicted the overall testing we would perform in Helsinki, and then present this test strategy/plan to the project managers.

As clarified throughout this thesis, I did not work alone in this project, since the complexity of it would have been much more than I could have handled by myself, however, I was part of the core team during the whole process of our testing in Helsinki, and my contributions were valuable to advance the technical quality of the E2EE solution and made the process of KMF's defect management smooth for the rest of the team, so they could concentrate in their own areas.

6. Comparison with other solutions

Secure Land Communications already had in its portfolio a solution to deploy End-to-End Encrypted communication, however, this solution fell short in terms of scalability and performance for wider networks, for this reason, the End-to-End Encryption solution described in this thesis was created, in order to solve a problem from one of our customers, the deployment in their network of End-to-End Encryption mechanisms for hundreds of thousands of users.

There are significant differences in the implementation of E2EE between the old solution, and the new solution. First, I will show a graphical overview of the old solution's architecture, after that, I will explain the main differences when comparing that old solution, with the new one described throughout this thesis, and at the end, I will mention End-to-End Encryption in systems other than TETRA.

6.1. Overview of our old solution for End-to-End Encryption

Following is a simplified example of the architecture comprising the KMC E2EE solution

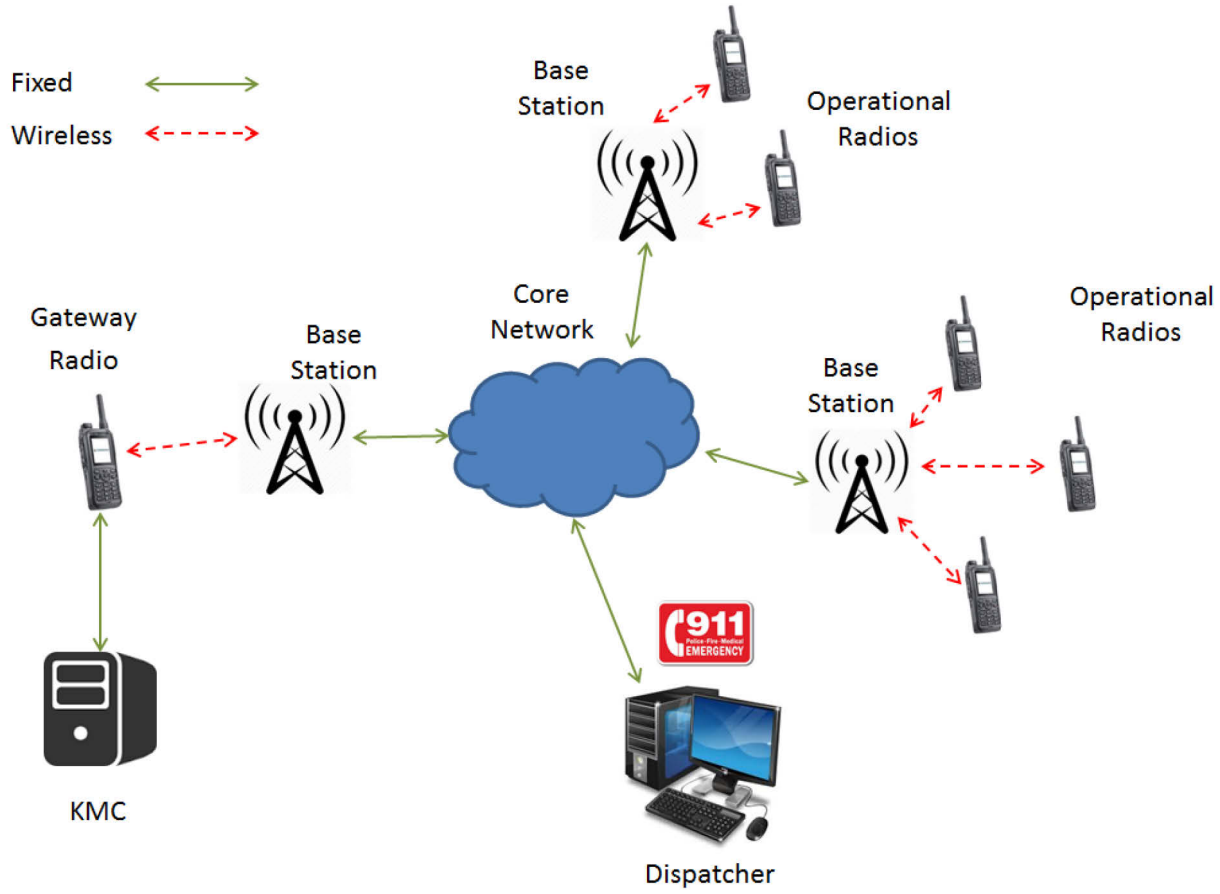

Figure 25 - KMC based E2EE solution architecture

As shown in figure 25, there is one dispatcher and 3 base stations connected to the core network. The dispatcher, together with the base stations containing the operational radios are working in a regular way, making and receiving calls, sending text message, and other daily operations. The base station on the left, however, is dedicated for one sole purpose, to connect the Key Management Center to the core network for exchange of encryption keys between itself, and the users (radios, dispatchers) to whom it serves. This is achieved by using a TETRA Radio as a gateway between the server and the base station.
All messages use the OTAK protocol, and Short Data Service (equivalent to SMS in GSM) is used as a bearer for this messages, just as in our solution featuring KMF.

The inherent nature of the radio air interface make a bottleneck of traffic in the network when a server tries to deliver content to the end users, and that makes this solution unsuitable when there is a large amount of users in the network.

6.2. Comparison between End-to-End Encryption featuring Key Management Center and Key Management Facility

As we notice from figure 25 and figure 21 from section 4.6.1.2, the main difference between the old and new solutions is the way the server connects to the core network.

In the old solution, KMC is inside of the TETRA network, connected to the exchange via a radio gateway, which in turn connects to a base station. The throughput on the air interface is quite low compared to a fixed network, and it very much limits the amount of data that can be sent/received by the server, for this reason, that solution was designed to support only about 500 subscribers.

The new solution comprising KMF is designed to support up to 40 000 users, the server, renamed Key Management Facility instead of Key Management Center, is acting a third party application outside the TETRA network, and connects over Ethernet via a API called TETRA Connectivity Server, the methods used in this API translate the instructions from the server to the TETRA network when sending/receiving data.
KMF is a completely different product in comparison with KMC, the theory and basis are the same, but the implementation is completely different, because KMF has been designed to support almost 100 times more users than its predecessor, for this reason, the internals in the server were designed to support a better performance, scalability and response times.

There are some other minor differences in between the old E2EE solution based on KMC, and the new one based in KMF, however, they are side effects of the new implementation and do not have a major importance to be mentioned here.

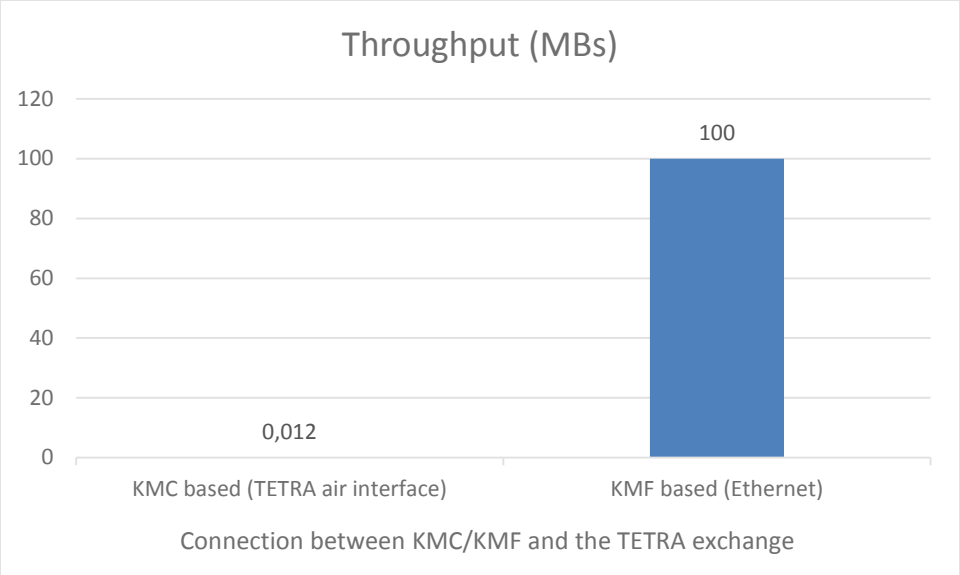

Figure 26.a - Throughput between KMC/KMF and DXT

The throughput mentioned in figure 26.a only takes into account the connection between the KMC/KMF and the Digital eXchange for TETRA (DXT), not the whole path from the KMC/KMF and the end user. The whole path is not included here because it's variable in nature, due to the fact that it depends on the Air Interface behavior between the end users/radios and the TETRA Base Stations where those radios are connected.

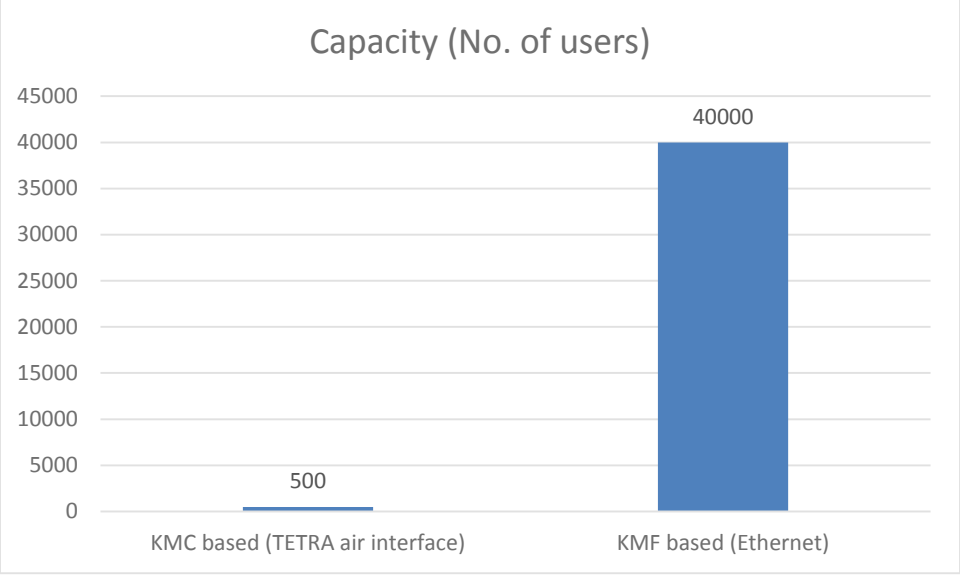

Figure 26.b - Number of users supported in KMC/KMF

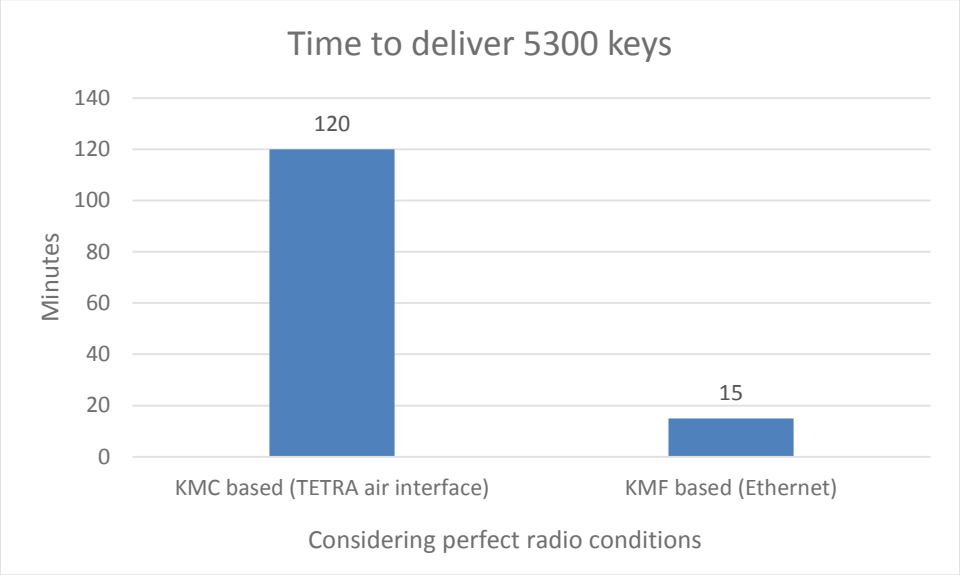

Figure 26.c - Delivery time for one crypto period's keys within one TETRA Base Station

Figure 26.c shows the average time to deliver 5300 keys from the KMC/KMF to the 35 TETRA Radios operating within one TETRA Base Station, this calculations take into account perfect radio conditions (no fading or background traffic).

In Figure 26.a (throughput), we see that the throughput comparison shows a much bigger difference compared to figure 26.c (delivery time). The smaller difference in figure 26.c (delivery) is due to the fact that it includes the whole path to deliver the keys, that is, when the testing was done, the OTAK protocol messages traverse the whole path from KMF, all the way through the Ethernet wired network, DXT, TBS and TETRA air interface. Being this last section the slowest point causing a bottleneck in the flow of the traffic.
In our new End-to-End Encryption solution, even if the connection between the server (KMF) and the exchange (DXT) uses Ethernet connectivity, the other side of the connection (air interface between base station and end user's radio) still uses a 9.6 Kbs circuit switched connection to send and receive Short Data Messages, hence, the difference in the times in figure 26.c is not as significant as the difference in throughput in figure 26.a.

6.3. Solutions in public commercial networks (non-TETRA)

At the moment of writing this thesis, there is a big boom of end-to-end encryption implementation in mainstream networks due to the recent deployment of end-to-end encryption by WhatsApp in its services. According to the magazine Fortune (2016), other chat service providers such as Viber are also following WhatsApp in the direction of end-to-end encryption.

WhatsApp End-to-End Encryption

As specified by WhatsApp (2016) in its end-to-end encryption overview technical paper, the company has starting to use the mentioned technology to encrypt all its conversations and file transfers (including any kind of media) for all its users.

There are few significant differences from WhatsApp approach to end-to-end encryption and our approach in Secure Land Communications:

	WhatsApp	SLC
Type of Encryption	Asymmetric	Symmetric
Encryption key life time	Per message	Per crypto period
Authentication	Always included in header	Performed by network
Session setup	Established once	No session needed
Message/Call speed	Slow	Very fast
IP packet data supported	Yes	No
Network involvement	Only relays messages	Can record traffic

Table 9 - Main differences between WhatsApp and SLC's E2EE systems

The reason for the differences shown in table 9 is due to the business needs in both public and private networks.
Since WhatsApp is an application running over public internet, it is available to the general public, on the other hand, TETRA networks are aimed mostly to public safety organizations such as police, army or fire departments, for this reason, the requirements and business needs are different in nature.

WhatsApp has a very good encryption mechanism which uses one encryption key per every message exchanged, this carries very good security, however, because it needs to derive a key every single time it sends content, it is very slow for the needs of our customers which require fast performance in emergency situations.

Another major difference is the role of the network infrastructure in the end-to-end encryption process. While in public commercial networks such as the internet, WhatsApp can provide true end-to-end

encryption by generating keys in the end user equipment when they install the application, that leaves the network servers in the role of a relay server, whose only task is to pass along the encrypted messages, this is an issue in the eyes of law enforcement agencies, as described by the New York Times (2016).
On the other hand, because our customers are mostly public security organisms, they require access to end-to-end encrypted material for lawful interception purposes, while at the same time they need end-to-end encryption to be protected in case parts of the networks get compromised. For this reason, this project's implementation of E2EE uses the KMF, in order to generate and distribute the keys to end user equipment, but also keeps a copy in a recorder server which can be used in case communication needs to be monitored.

6.4. How to apply this solution in public mobile services/technologies

It is somewhat hard to visualize this solution applied to the mainstream mobile communications for the following reasons:

a) In public mobile networks such as GSM/UMTS/LTE, there is no need to divide the users in organizations. Everybody can speak to any other person in earth as long as there is connectivity between their carriers; users are not limited to certain groups or areas. In this way, the concept of organizations and groups/fleets is no longer valid, which makes control encryption keys more difficult.
b) In private networks such as TETRA, the amount of users is not so high, plus the "organizations" concept limits the amount of users to only a couple of thousands, which can easily be served by the KMF. In public networks, 40 000 users (the amount supported by KMF) wouldn't be even close to the amount needed to support end-to-end encryption in a country.

On the other hand, if the following requisites are satisfied, the end-to-end encryption system explained in the thesis would be applicable:

a) There is a need/law in a country mandating the possibility to record communications for lawful interception, while at the same time keeping confidentiality by encrypting communications for eavesdroppers/hackers.
b) If the amount of users is too high, replication of certain users' keys for inter-city communication is performed among KMFs in order to accomplished long distance communication.
c) If a company or institution has non-TETRA mobile technology, such as UMTS, the same concepts for encryption may be applied, as long as the KMF server is coded in a way that instead of sending OTAK protocol messages for a TETRA API, it sends messages designed to interact with the mentioned mobile technology (UMTS in this example).

7. Summary and conclusions

This Thesis has served the purpose of finding a number of significant errors mostly in the core product of the solution, the Key Management Facility.

Due to the fact that the KMF is newly developed, critical and interesting errors have been found during the testing phase we have performed in Helsinki. Which, if not found, would have caused severe consequences in the customer's network, which would have led to not only the misbehavior of other devices and bad performance, but also affect the image of Secure Land Communication's products in terms of quality.

By performing the testing described in this thesis, we have ensured that adding to the customer's network the Ent-to-End Encryption solution with Over The Air Keying capability will result in a significant reduction of management efforts for the customer's technical personnel in terms of delivery of encryption keys to the TETRA Radio equipment present in the network, this without the risk of unwanted side effects.

Both the TETRA Radios (not covered in this thesis) and the Key Management Facility server were significantly improved during the testing efforts performed throughout this thesis work. However, the major benefit of this thesis' work, in my opinion, is the addition of a new product (and solution overall) to the company's portfolio, which will yield benefits not only now that it will be deployed with our current customer, but also in the future since we will be able to offer it to other customers as a neat feature of our systems.

It is true that the work during this thesis did not solve all the problems, however, it improved the solution's quality significantly. In the next section, I will discuss what is left and needs to be addressed in the future.

8. Future work

Although most critical issues have been solved until the time of writing this thesis, there are still non-critical issues which must be corrected in subsequence versions of the solution.

For instance, the KMF's GUI is a weak point in the overall solution, mostly because of the inflexibility of modifying some parameters, as for example, the ability to delete old Key Encryption Keys from KMF's database and other small details which do not cause a direct impact on the operations of the network/solution, but which can be noticed by the customer once he starts using the system.

Another area of improvement could be to use talk group addressing (Group Short Subscriber Identity) instead of individual addressing (Individual Short Subscriber Identity) to deliver the OTAK messages, this would very much benefit the impact of the OTAK traffic on the downlink direction, however, a technique to acknowledge correctly the reception of the mentioned OTAKs must be designed.

References

Bishop, M. (2008) Introduction to Computer Security. 4th Ed. Westford, Massachusetts: Addison-Wesley.

ETSI (2016) TETRA. Available at: http://www.etsi.org/technologies-clusters/technologies/tetra [Accessed: 29th February 2016]

Finland. Airbus Defence and Space. (2015, a). TETRA E2EE COURSE OVERVIEW (KMF Based E2EE solution using OTAK).

Finland. Airbus Defence and Space (2015, b). Product Description of the TETRA Dispatcher Workstation (DWS). Finland: Airbus Defence and Space. (TETRA System Release 6.5 – 7, DN05221844-10-0en).

Finland. Airbus Defence and Space (2015, c). TCS Product Description. Finland: Airbus Defence and Space. (TETRA System Release 6.0-7.0, DN0116031-20-0en).

Finland. Airbus Defence and Space (2015, d). Training course. Finland: Airbus Defence and Space. (TETRA System Course, ED07.01en).

Finland. Cassidian (2013). Authentication in the TETRA System. Finland: Cassidian. An EADS Company. (TETRA SYSTEM RELEASE 6.0 – 6.5, TRASYSAPP00003-02-3en).

Fortune (2016) Another Big Messaging App Joins the End-to-End Encryption Party. Available at: http://fortune.com/2016/04/19/viber-e2e-encryption/ [Accessed: 27th April 2016]

Ketterling, H. (2003) Introduction to Digital Professional Mobile Radio. Norwood, US: Artech House Books.

New York Times (2016) WhatsApp Introduces End-to-End Encryption. Available at: http://www.nytimes.com/2016/04/06/technology/whatsapp-messaging-service-introduces-full-encryption.html?_r=0 [Accessed: 27th April 2016]

Poole, I. (2006) Cellular Communications Explained : From Basics to 3G. GBR: Newnes: Jordan Hill.

Secure Land Communications (2015) Solutions [online] Available from: http://www.securehybridcomms.com/solutions [Accessed: 13th October 2015]

SELEX communications (2007) Introduction to TETRA. Available at: http://www.ok1mjo.com/all/tetra/vseobecne_informace/TETRA_Selex_Introduction_to_TETRA_TErrestrial-Trunked-RAdio.pdf [Accessed: 29th February 2016]

Stallings, W. & Brown, L. (2008) Computer Security Principles and Practice. Upper Saddle River: Pearson Prentice Hall.

Sturtzel, A. (2015) Airbus Defence and Space unveils new DXTA Tetra server with higher capacity and new capabilities. Available at: http://www.airbusgroup.com/int/en/news-media/press-releases/Airbus-Group/Financial_Communication/2015/05/20150520_airbus_defence_and_space_dxta_tetra_server.html [Accessed: 17th February 2016]

TCCA (2016) TETRA Security. Available at: http://www.tandcca.com/about/page/12027 [Accessed: 2nd March 2016]

WhatsApp (2016) WhatsApp Encryption Overview Technical Whitepaper. Available at: https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf [Accessed: 27th April 2016]

List of Figures

List of Tables